UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA/INFORMÁTICA
CURSO DE ENGENHARIA DA COMPUTAÇÃO

RELATÓRIO FINAL

OCILOSCÓPIO DIGITAL DE BAIXO CUSTO

TIAGO COELHO
GEOVANE FERREIRA
VINICIUS ZARAMELLA
HENRIQUE SARMENTO

CURITIBA
2011

TIAGO COELHO
GEOVANE FERREIRA
VINICIUS ZARAMELLA
HENRIQUE SARMENTO

OCILOSCÓPIO DIGITAL DE BAIXO CUSTO

Relatório final apresentada como requisito para aprovação na disciplina de Oficina de Integração 3, ministrada pelos professores:

Prof. Dr. João Alberto Fabro

Prof. Dr. Heitor Silvério Lopes

CURITIBA
2011

RESUMO

FERREIRA, Geovane; COELHO, Tiago; ZARAMELLA, Vinicius; SARMENTO, Henrique. Osciloscópio Digital de Baixo Custo. Relatório Final – Curso de Engenharia da Computação, Universidade Tecnológica Federal do Paraná. Curitiba, 2011

Este projeto tem como objetivo a montagem de um osciloscópio digital de baixo custo, provendo aos estudantes de engenharia eletrônica e da computação uma ferramenta acessível. O projeto inclui um sistema de aquisição de sinais que condiciona o sinal para faixas em que o ADC trabalha. Os dados analógicos são então convertidos em digitais e enviados do sistema embarcado para a estação-base onde são plotados em software.

Palavras-chave

Osciloscópio Digital, Condicionamento de Sinal, Sistema de Aquisição de Dados.

ABSTRACT

FERREIRA, Geovane; COELHO, Tiago; ZARAMELLA, Vinicius; SARMENTO, Henrique. Osciloscópio Digital de Baixo Custo*,* Low Cost Digital Oscilloscope. Final Report - Computer Engineering, Universidade Tecnológica Federal do Paraná. Curitiba, 2011.

This project aims at setting up a low-cost digital oscilloscope, providing students of electrical engineering and computing engineering a handy tool. The project includes a signal acquisition system that conditions the signal to the ADC tracks that works. The analog data is then converted to digital and sent to the embedded system where in the base station are plotted in software.

Key-words

Digital Oscilloscope, Signal Conditioning, Data Acquisition System.

Lista de Tabelas

Tabela 1 - Modelos do PicoScope 15
Tabela 2 - Componentes que atendem os requisitos do projeto. 27
Tabela 3 - Componentes que não atendem os requisitos do projeto. 28
Tabela 4 - ADCs Pesquisados 28
Tabela 5 - Opções pesquisadas de fonte chaveada 29
Tabela 6 - Amplificadores pesquisados 29
Tabela 7 - Comparações linguagens de programação 30
Tabela 8 - Orçamento detalhado previsto 34
Tabela 9 - Deliverables e auxiliar de gerente 37

Lista de Figuras

Figura 1 – Cronograma 35
Figura 2 - Mapa de Gantt 36
Figura 3 - Diagrama de blocos osciloscópio 39
Figura 4 - Painel frontal de um osciloscópio digital 40
Figura 5 - Diagrama de blocos do STM32 43
Figura 6 - Encapsulamente LQFP-64 STM32F103 48
Figura 7 - kit de desenvolvimento STM-P103 49
Figura 8 - ST-LINK V2 da STMicroeletronics 50
Figura 9 - Visão geral em diagrama de blocos 53
Figura 10 - Sistema de aquisição de sinal 54
Figura 11 - Sistema de alimentação 54
Figura 12 - Circuito de entrada do primeiro canal 55
Figura 13 - Circuito de entrada do segundo canal 55
Figura 14 - Pinos de conexão com a placa de desenvolvimento 55
Figura 15 - Pacote de Comando 57
Figura 16 – Pacote de dados 57
Figura 17 – Pacote de reconhecimento 58
Figura 18 – Protocolo de comunicação estação base 59
Figura 19 - Protocolo de comunicação sistema embarcado 60
Figura 20 – Diagrama de Máquina de Estados do Sistema Embarcado 61
Figura 21 - Diagrama 2 de Máquina de Estados do Sistema Embarcado 62
Figura 22 - Diagrama de casos de uso 64
Figura 23 - Diagrama de Classes 66
Figura 24 – Diagrama de sequência para alterar escala de tempo do canal 1 68
Figura 25 – Diagrama de sequência para alterar escala de tesão do canal 1 68
Figura 26 – Diagrama de sequência para mostrar dados dos cursores 69
Figura 27 – Diagrama de sequência para apresentar dados dos canais 69
Figura 28 – Diagrama de sequência para apresentar o gráfico dos canais 70
Figura 29 – Diagrama de sequência para ativar cursores 70
Figura 30 – Diagrama de sequência para ativar o trigger 71
Figura 31 – Diagrama de sequência para desativar os cursores 71
Figura 32 – Diagrama de sequência para desativar o trigger 72
Figura 33 – Diagrama de sequência para deslocar os cursores 72
Figura 34 – Diagrama de sequência para deslocar o trigger 73
Figura 35 – Diagrama de sequência para iniciar conexão 73
Figura 36 – Diagrama de sequência para eventos Run e Stop 74

Figura 37 – Diagrama de sequência para selecionar o Anti-Aliasing no canal1 74
Figura 38 – Diagrama de sequência para desativar o Anti-Aliasing do canal1 74
Figura 39 - Face superior da PCI do SAD 76
Figura 40 - Face inferior da PCI do SAD 77
Figura 41 - Face superior da PCI do microcontrolador 78
Figura 42 - Face inferior da PCI do microcontrolador 78
Figura 43 - Gerador de Funções ATF20B 80
Figura 44 - Osciloscópio DSO3062A 81
Figura 45 - Fonte de Alimentação do Departamento de Eletrônica da UTFPR 81
Figura 46 - Código main do software do microcontrolador 82
Figura 47 - DMA Handler 83
Figura 48 - Configurações dos clocks 85
Figura 49 - Configuração GPIO 86
Figura 50 - Configuração GPIO segunda parte 87
Figura 51 - Configuração ADC 89
Figura 52 - Configuração DMA 90
Figura 53 - Configuração timer 91
Figura 54 - Configuração Interrupções 92
Figura 55 - Gerberer de furos do SAD 94
Figura 56 - Gerberer de trilhas do SAD 95
Figura 57 - Divisor puramente resistivo 97
Figura 58 - Divisor com capacitância intrínseca 97
Figura 59 - Divisor resistivo elaborado para o projeto 98
Figura 60 – Formula para o calculo da compensação 98
Figura 61 - Tensão RMS através de uma série de pontos 103
Figura 62 - Interface gráfica software 106
Figura 63 – PCI do sistema de aquisição de sinais 107
Figura 64 – PCI do sistema de aquisição de sinais acoplada ao kit de desenvolvimento 108
Figura 65 - Osciloscópio projetado [Vpp = 20,664 V; Vrms = 7,288 V; f = 0,986 Hz] 109
Figura 66 - Osciloscópio comercial [Vpp = 20,2 V; Vrms = 7 V; f = 1 Hz] 110
Figura 67 - Osciloscópio projetado [Vpp = 121,867 mV; Vrms = 38,312 mV; f = 992,056 Hz] 110
Figura 68 - Osciloscópio comercial [Vpp = 118 mV; Vrms = 36,5 mV; f = 1 kHz] 111
Figura 69 - Osciloscópio projetado [Vpp = 126,294 mV; Vrms = 53,003 mV; f = 992,056 Hz] 111
Figura 70 - Osciloscópio comercial [Vpp = 123 mV; Vrms = 52,8 mV; f = 1 kHz] 112
Figura 71 - Osciloscópio projetado [Vpp = 1,134 V; Vrms = 397,669 mV; f = 96,154 kHz] 112
Figura 72 - Osciloscópio comercial [Vpp = 1,07 V; Vrms = 359 mV; f = 101 kHz] 113
Figura 73 - Osciloscópio projetado [Vpp = 2,047 V; Vrms = 0,976 V; f = 1 Hz] 113
Figura 74 - Osciloscópio comercial [Vpp = 2,08 V; Vrms = 1,02 V; f = 1 Hz] 114
Figura 75 - Osciloscópio projetado [Vpp = 2,381 V; Vrms = 0,970 V; f = 9,978 kHz] 114
Figura 76 - Osciloscópio comercial [Vpp = 2,08 V; Vrms = 1,02 V; f = 10 kHz] 115
Figura 77 - Osciloscópio projetado [Vpp = 2,361 V; Vrms = 0,941 V; f = 19,956 kHz] 115
Figura 78 - Osciloscópio comercial [Vpp = 2,08 V; Vrms = 1,01 V; f = 20 kHz] 116
Figura 79 - Osciloscópio projetado [Vpp = 2,124 V; Vrms = 0,752 V; f = 96,154 kHz] 116
Figura 80 - Osciloscópio comercial [Vpp = 2,12 V; Vrms = 1,01 V; f = 100 kHz] 117
Figura 81 - Osciloscópio projetado [Vpp = 440,146 mV; Vrms = 156,605 mV; f = 156,250 kHz] 117
Figura 82 - Osciloscópio comercial [Vpp = 2,12 V; Vrms = 1,01 V; f = 150 kHz] 118
Figura 83 - Osciloscópio projetado [Vpp = 103,44 mV; Vrms = 57,546 mV; f = 62,500 kHz] 119
Figura 84 - Osciloscópio comercial [Vpp = 2,12 V; Vrms = 1,01 V; f = 200 kHz] 119
Figura 85 - Resposta em frequência do anti-alising 121
Figura 86 - Cronograma atualizado 122
Figura 87 - Gantt atualizado 123
Figura 88 - Osciloscópio projetado [Vpp = 18,493 V; Vrms = 6,484 V; f = 9,978 kHz] 131
Figura 89 - Osciloscópio comercial [Vpp = 20,4 V; Vrms = 6,99 V; f = 10 kHz] 131
Figura 90 – Osciloscópio projetado [Vpp = 15,252 V; Vrms = 5,364 V; f = 992,056 Hz] 132
Figura 91 - Osciloscópio comercial [Vpp = 15,2 V; Vrms = 5,25 V; f = 1kHz] 132
Figura 92 - Osciloscópio projetado [Vpp = 15,643 V; Vrms = 5,507 V; f = 0,986Hz] 133
Figura 93 - Osciloscópio comercial [Vpp = 15,4 V; Vrms = 5,28 V; f = 1,01Hz] 133
Figura 94 - Osciloscópio projetado [Vpp = 13,899 V; Vrms = 4,899 V; f = 10,210 kHz] 134
Figura 95 - Osciloscópio comercial [Vpp = 15,2 V; Vrms = 5,24 V; f = 10 kHz] 134
Figura 96 - Osciloscópio projetado [Vpp = 13,693 V; Vrms = 4,853 V; f = 96,154 kHz] 135
Figura 97 - Osciloscópio comercial [Vpp = 15,1 V; Vrms = 5,21V; f = 100 kHz] 135
Figura 98 - Osciloscópio projetado [Vpp = 14,681 V; Vrms = 6,593 V; f = 96,154 kHz] 136
Figura 99 - Osciloscópio comercial [Vpp = 15,7 V; Vrms = 7,57 V; f = 100 kHz] 136
Figura 100 - Osciloscópio projetado [Vpp = 14,398 V; Vrms = 6,948 V; f = 9,978 kHz] 137
Figura 101 - Osciloscópio comercial [Vpp = 15,4 V; Vrms = 7,59 V; f = 10 kHz] 137
Figura 102 - Osciloscópio projetado [Vpp = 15,705 V; Vrms = 7,538 V; f = 992,056 Hz] 138

Figura 103 - Osciloscópio comercial [Vpp = 15,5 V; Vrms = 7,6 V; f = 1 kHz] 138
Figura 104 - Osciloscópio projetado [Vpp = 15,725 V; Vrms = 7,784 V; f = 0,986 Hz] 139
Figura 105 - Osciloscópio comercial [Vpp = 15,3 V; Vrms = 7,55 V; f = 1,01 Hz] 139
Figura 106 - Osciloscópio projetado [Vpp = 2,047 V; Vrms = 0,976 V; f = 1 Hz] 140
Figura 107 - Osciloscópio comercial [Vpp = 2,08 V; Vrms = 1,02 V; f = 1 Hz] 140
Figura 108 - Osciloscópio projetado [Vpp = 2,134 V; Vrms = 0,978 V; f = 992,056 Hz] 141
Figura 109 - Osciloscópio comercial [Vpp = 2,08 V; Vrms = 1,02 V; f = 1 kHz] 141
Figura 110 - Osciloscópio projetado [Vpp = 2,126 V; Vrms = 0,998 V; f = 9,978 kHz] 142
Figura 111 - Osciloscópio comercial [Vpp = 2,08 V; Vrms = 1,02 V; f = 10 kHz] 142
Figura 112 - Osciloscópio projetado [Vpp = 2,128 V; Vrms = 0,980 V; f = 96,154 kHz] 143
Figura 113 - Osciloscópio comercial [Vpp = 2,14 V; Vrms = 1,01 V; f = 100 kHz] 143
Figura 114 - Osciloscópio projetado [Vpp = 2,049 V; Vrms = 0,726 V; f = 104,167 kHz] 144
Figura 115 - Osciloscópio comercial [Vpp = 2,04 V; Vrms = 0,696 V; f = 100 kHz] 144
Figura 116 - Osciloscópio projetado [Vpp = 2,016 V; Vrms = 0,704 V; f = 9,978 kHz] 145
Figura 117 - Osciloscópio comercial [Vpp = 2,06 V; Vrms = 0,701 V; f = 10 kHz] 145
Figura 118 - Osciloscópio projetado [Vpp = 1,965 V; Vrms = 0,685 V; f = 992,056 Hz] 146
Figura 119 - Osciloscópio comercial [Vpp = 2,04 V; Vrms = 0,702 V; f = 1 kHz] 146
Figura 120 - Osciloscópio projetado [Vpp = 1,969 V; Vrms = 0,704 V; f = 1 Hz] 147
Figura 121 - Osciloscópio comercial [Vpp = 2,04 V; Vrms = 0,718 V; f = 1,01 Hz] 147
Figura 122 - Osciloscópio projetado [Vpp = 1,037 V; Vrms = 367,088 mV; f = 1 Hz] 148
Figura 123 - Osciloscópio comercial [Vpp = 1,05 V; Vrms = 369 mV; f = 1 Hz] 148
Figura 124 - Osciloscópio projetado[Vpp = 1,055V; Vrms = 366,066mV; f = 992,056Hz] 149
Figura 125 - Osciloscópio comercial [Vpp = 1,09 V; Vrms = 366 mV; f = 1 kHz] 149
Figura 126 - Osciloscópio projetado [Vpp = 1,119 V; Vrms = 386,653 mV; f = 9,978 kHz] 150
Figura 127 - Osciloscópio comercial [Vpp = 1,08 V; Vrms = 364 mV; f = 10 kHz] 150
Figura 128 - Osciloscópio projetado [Vpp = 1,134 V; Vrms = 397,669 mV; f = 96,154 kHz] 151
Figura 129 - Osciloscópio comercial [Vpp = 1,07 V; Vrms = 359 mV; f = 101 kHz] 151
Figura 130 - Osciloscópio projetado [Vpp = 1,189 V; Vrms = 0,545 V; f = 96,154 kHz] 152
Figura 131 - Osciloscópio comercial [Vpp = 1,1 V; Vrms = 0,520 V; f = 100 kHz] 152
Figura 132 - Osciloscópio projetado [Vpp = 1,202 V; Vrms = 0,552 V; f = 9,978 kHz] 153
Figura 133 - Osciloscópio comercial [Vpp = 1,1 V; Vrms = 0,526 V; f = 10 kHz] 153
Figura 134 - Osciloscópio projetado [Vpp = 1,232 V; Vrms = 0,527 V; f = 1,05 kHz] 154
Figura 135 - Osciloscópio comercial [Vpp = 1,1 V; Vrms = 0,530 V; f = 1 kHz] 154
Figura 136 - Osciloscópio projetado [Vpp = 1,075 V; Vrms = 0,516 V; f = 1 Hz] 155
Figura 137 - Osciloscópio comercial [Vpp = 1,09 V; Vrms = 0,517 V; f = 1 Hz] 155
Figura 138 - Osciloscópio projetado [Vpp = 124,471 mV; Vrms = 38,837 mV; f = 1 Hz] 156
Figura 139 - Osciloscópio comercial [Vpp = 126 mV; Vrms = 36,3 mV; f = 0,994 Hz] 156
Figura 140 - Osciloscópio projetado [Vpp = 121,607 mV; Vrms = 38,323 mV; f = 9,978 kHz] 157
Figura 141 - Osciloscópio comercial [Vpp = 117 mV; Vrms = 36,3 mV; f = 10 kHz] 157
Figura 142 - Osciloscópio projetado [Vpp = 127,075 mV; Vrms = 42,178 mV; f = 104,167 kHz] 158
Figura 143 - Osciloscópio comercial [Vpp = 112 mV; Vrms = 35,6 mV; f = 100 kHz] 158
Figura 144 - Osciloscópio projetado [Vpp = 117,701mV; Vrms = 51,953 mV; f = 96,154 kHz] 159
Figura 145 - Osciloscópio comercial [Vpp = 119 mV; Vrms = 51,5 mV; f = 100 kHz] 159
Figura 146 - Osciloscópio projetado [Vpp = 128,117 mV; Vrms = 53,332 mV; f = 9,978 kHz] 160
Figura 147 - Osciloscópio comercial [Vpp = 121 mV; Vrms = 52,3 mV; f = 10 kHz] 160
Figura 148 - Osciloscópio projetado [Vpp = 126,294 mV; Vrms = 53,003 mV; f = 992,056 Hz] 161
Figura 149 - Osciloscópio comercial [Vpp = 123 mV; Vrms = 52,8 mV; f = 1 kHz] 161
Figura 150 - Osciloscópio projetado [Vpp = 130,461 mV; Vrms = 53,433 mV; f = 0,989 Hz] 162
Figura 151 - Osciloscópio comercial [Vpp = 122 mV; Vrms = 51,3 mV; f = 1 Hz] 162
Figura 152 - Software osciloscópio 163

Sumário

1 INTRODUÇÃO 11
1.1 MOTIVAÇÃO 11
1.2 OBJETIVOS 11

2 PLANEJAMENTO 12
2.1 PREMISSAS E RESTRIÇÕES 12
2.2 DESIGNAÇÃO DO GERENTE DA EQUIPE 12
2.3 REQUISITOS 12
2.3.1 Requisitos funcionais 12
2.4 TRABALHOS CORRELATOS 13
2.4.1 DSO-Nano 13
2.4.2 PicoScope 14
2.4.3 BitScope 15
2.5 PLANEJAMENTO DE RISCOS 16
2.6 ALTERNATIVA TECNOLÓGICA 26
2.6.1 HARDWARE 26
2.6.1.1 Microcontrolador 26
2.6.1.2 ADC 28
2.6.1.3 Charge Pump (Fonte Chaveada) 28
2.6.1.4 Amplificadores 29
2.6.2 SOFTWARE 30
2.6.2.1 Linguagem de Programação de Estação Base 30
2.6.2.2 Sistema Operacional da Estação Base 31
2.6.2.3 Ambiente integrado para o desenvolvimento do software da Estação Base 31
2.6.2.4 Programador e Debugger 32
2.6.2.5 Ambiente integrado para o desenvolvimento do software do microcontrolador 32
2.6.2.6 Linguagem de Programação do microcontrolador. 32
2.7 ORÇAMENTO INICIAL DETALHADO PREVISTO 33
2.8 ORÇAMENTO DETALHADO 34
2.9 CRONOGRAMA 34
2.10 ENTREGÁVEIS 37

3 FUNDAMENTAÇÃO TEÓRICA 39
3.1 OSCILOSCÓPIO 39
3.1.1 Funcionalidades 40
3.1.2 Condicionamento de Sinais 41
3.1.3 Amostragem de Sinais Variantes no Tempo 41
3.1.3.1 Limite de Nyquist 41
3.2 STM32F103 42
3.2.1 Core 43
3.2.2 Periféricos 45
3.2.2.1 ADC 45
3.2.2.2 Timers 45
3.2.2.3 DMA 46
3.2.2.4 JTAG 47
3.2.3 Encapsulamento 48
3.2.4 Kit de Desenvolvimento 48
3.2.5 Programador/Debugger 50
3.3 SISTEMA DE COMUNICAÇÃO 50
3.3.1 STM32F103– Firmware 50
3.3.2 Estação-Base- API javax.comm 51
3.4 JFREECHART 51

4 PROJETO 53

4.1 DIAGRAMA DE BLOCOS DO SISTEMA 53
4.2 CIRCUITO DE CONDICIONAMENTO DE SINAL 53
4.2.1 Diagrama Esquemático 53
4.3 PROTOCOLO DE COMUNICAÇÃO 56
4.3.1 Diagrama de Máquina de Estados 58
4.3.1.1 Estação Base 59
4.3.1.2 Sistema Embarcado 59
4.4 SISTEMA EMBARCADO 60
4.4.1 Diagrama de Máquina de Estados 60
4.5 ESTAÇÃO BASE 63
4.5.1 Diagrama de Casos de Uso 63
4.5.2 Diagrama de Classes 66
4.5.3 Diagramas de Sequência 68
4.6 PLACA DE CIRCUITO IMPRESSO 75
4.6.1 Placa do Circuito de Condicionamento de Sinal 75
4.6.2 Placa do Circuito de Condicionamento de Sinal com Microcontrolador Integrado 77

5 EXECUÇÃO 79
5.1 COMPONENTES UTILIZADOS 79
5.2 EQUIPAMENTOS UTILIZADOS 80
5.3 SISTEMA EMBARCARCADO 82
5.3.1 Configuração dos periféricos 84
5.3.1.1 Clock do core e dos periféricos 84
5.3.1.2 GPIO 86
5.3.1.3 ADC 88
5.3.1.4 DMA 89
5.3.1.5 Timer 90
5.3.1.6 Interrupções 91
5.3.2 Calibragem da taxa de amostragem 92
5.3.3 Dificuldades encontradas 93
5.4 CONDICIONAMENTO DE SINAL 93
5.4.1 Calibragem do circuito de condicionamento de sinal 93
5.4.2 Confecção da PCI 93
5.4.3 Dificuldades encontradas 95
5.5 COMUNICAÇÃO 99
5.5.1 Implementação do sistema embarcado 99
5.5.2 Implementação da estação base 99
5.5.3 Dificuldades encontradas 100
5.6 ESTAÇÃO BASE 101
5.6.1 Método de plotagem 101
5.6.2 Medições 102
5.6.2.1 frequência 102
5.6.2.2 Tensão pico a pico 102
5.6.2.3 Tensão Rms 102
5.6.3 Cursores 103
5.6.4 Trigger 103
5.6.5 Stop 103
5.6.6 Simulação de sinais para a estação base 104
5.6.7 Dificuldades encontradas 104

6 RESULTADOS 106
6.1 INTERFACE GRÁFICA DO OSCILOSCÓPIO 106
6.2 ANÁLISE DA PLACA DE CIRCUITO IMPRESSO 106
6.2.1 Comparação com Outros Osciloscópios 109

6.4.2 Resposta em frequência do anti-alising *120*

7 CRONOGRAMA **122**

8 CONCLUSÕES **126**

8.1 ANÁLISE DO DESENVOLVIMENTO 126

8.2 INTEGRAÇÃO 127

8.3 TRABALHOS FUTUROS 128

9 ANEXOS **131**

9.1 COMPARAÇÃO COM OUTROS OSCILOSCÓPIOS 131

9.2 MANUAL DO USUÁRIO 163

10 REFERÊNCIAS **165**

1 Introdução

1.1 MOTIVAÇÃO

A escolha por esse projeto foi motivada inicialmente pelos próprios integrantes da equipe necessitarem de um equipamento que meça tensões de sinais analógicos ou digitais, em experimentos de matérias no curso de engenharia de computação como Fundamentos De Análise De Circuitos Elétricos, Eletrônica Geral 1.

Em alguns laboratórios dessas matérias precisamos levantar a resposta em frequência, ou observar o formato da na onda na saída de um determinado circuito, para isso utilizamos um osciloscópio, geralmente o estudante precisa vir para a faculdade fazer o experimento por não ter determinados equipamentos, com o osciloscópio proposto neste projeto o estudante poderá analisar circuitos com sua ferramenta particular, necessitando somente de um computador.

1.2 OBJETIVOS

O objetivo do projeto é o desenvolvimento de um osciloscópio digital de baixo custo que opere em faixas de frequência de até 100KHz, o sistema desenhara os sinais de onda na estação base sendo necessária uma comunicação estação base – sistema embarcado. Além disso, será implementado um sistema de condicionamento de sinal que faz o ajuste de tensão para a faixa de funcionamento do ADC.

2 Planejamento

2.1 PREMISSAS E RESTRIÇÕES

PREMISSAS:

- Osciloscópio Digital
- Gerador de Funções
- Estação de solda
- Computador com entrada USB 2.0.

RESTRIÇÕES:

- Não garantimos a confiabilidade do sinal acima de 100kHz.
- Não garantimos a confiabilidade do sinal para tensão menor que 10 mV.
- Não garantimos a segurança do equipamento caso o usuário utilize um sinal superior a 20 V ou inferior a -20 V

2.2 DESIGNAÇÃO DO GERENTE DA EQUIPE

Gerente: Vinicius Zaramella
Colaboradores:

- Geovane Vinicius Ferreira
- Henrique Reinaldo Sarmento
- Tiago Rafael Volkmann Coelho

2.3 REQUISITOS

2.3.1 Requisitos funcionais

Requisitos do sistema:

RF01 – O sistema deve ser de baixo custo.

RF02 – O sistema deve possuir dois canais.

RF03 – Os dados convertidos devem possuir resolução de no mínimo 10 bits.

RF04 - O sinal amostrado deverá ter amplitude maior que 10 mV em relação ao *ground.*

RF05 – O circuito de entrada deve possuir filtro anti-aliasing que pode ser selecionado ou não.

RF06 – A comunicação deve ser USB

RF07 - A alimentação do sistema deve ser via USB

Os requisitos do Software são listado abaixo:

RF01 – O software deverá plotar os dados recebidos;

RF02 – O software deverá ter a função stop;

RF03 – O software deverá mostrar o estado da conexão USB;

RF04 – O software deverá ter plotar dados de dois canais;

RF05 – O software deverá ter a função de trigger;

RF06 – O software deverá mostrar a tensão RMS dos dois canais;

RF07 – O software deverá mostrar a frequência em Hz nos dois canais;

RF08 – O software deverá mostrar tensão pico à pico dos dois canais;

RF09 – O software deverá ter offset para os dois canais;

RF10 – O software deverá ter a função AntAliasing para os dois canais;

RF11 – O software deverá ter a função SingleShot;

RF12 – O software deverá ter escalas diversas de base de tempo;

RF13 - O software deverá ter escalas diversas de tensão para os dois canais;

RF14 – O software deverá ter dois cursores verticais;

RF15 – O software deverá apresentar os dados dos cursores relativos aos canais;

2.4 TRABALHOS CORRELATOS

2.4.1 DSO-Nano

Compreende em um osciloscópio digital compacto que foi projetado para exercer atividades básicas de engenharia, sendo assim contem baixa taxa de amostragem 1 MSPS e apenas 1 canal.

Diante do exposto o projeto desenvolvido foi baseado em uma plataforma ARM Cortex M3, possuindo ainda um display LCD colorido (320x240) onde o sinal é exibido, bem como contem uma entrada microSD card, conexão USB, bateria recarregável de maneira integrada. O custo do DSO-nano encontra-se na faixa de R$ 300.[1]

A principal vantagem do projeto proposto em relação ao DSO-nano é a visualização simultânea de dois canais, e a desvantagem é que nosso projeto necessita da estação-base para a visualização do sinal.

2.4.2 PicoScope

A empresa Pico Technology, desde 1991, vem investindo na produção de osciloscópio com baixo custo. O sinal captado é transmitido pela USB ou Ethernet (dependendo do modelo) sendo exibido em um PC com plataforma Windows.[2]

Logo abaixo uma tabela contendo os modelos do PicoScope existentes:

Modelo	Canais	Largura de banda	Amostragem	Resolução	Preço
2100	1	25MHz	100MS/s	8 Bits	US$ 328,35
2200	2	25MHz	200 MS/s	8 Bits	US$ 486,75
3000	2	200 MHz	500 MS/s	8 Bits	US$ 1.483,35
4224	2	20 MHz	80 MS/s	12 Bits	US$ 856,00
4224IEPE	2	20 MHz	80 MS/s	12 Bits	US$ 988,35
4226/4227	2	100 MHz	250 MS/s	12 Bits	US$ 1.483,35
4262	2	5 MHz	10 MS/s	16 Bits	US$ 1.235,85
4424	4	20 MHz	80 MS/s	12 Bits	US$ 1.318,35
3425	4	5 MHz	20 MS/s	12 Bits	US$ 1.608.75
5200	2	250 MHz	1 GS/s	8 Bits	US$ 2.961,75
6400	4	500 MHz	5 GS/s	8 Bits	US$

					9.891,75

Tabela 1 - Modelos do PicoScope

O osciloscópio mais barato da PicoScope possui somente um canal para analise de sinais. Nosso projeto terá a visualização de dois canais e com custo inferior ao modelo 2100 da PicoScope, a resolução será de 12 bits também. A largura de banda do modelo 2100, entretanto é maior que a proposta neste projeto.

2.4.3 BitScope

Trata - se de um osciloscópio que tem possibilidade de captar, simultaneamente, de 2 até 8 canais, enviando dados para um PC pelo cabo USB ou Ethernet.[3]

Os softwares do BitScop dispõe de grande variedade de recursos, sendo considerados aprimorados. O BitScop foi desenvolvido em três modelos, alguns tem a capacidade de gerar ondas de forma arbitrária (AWG).[3]

O preço do osciloscópio mais barato da Bitscope é de R$ 893,00, possuindo 2 canais de entrada, resolução de 8 bits, e largura de banda de 100MHz. As vantagens do nosso projeto em relação ao BitScope é o custo inferior e a resolução de 12 bits. Porém o BitScope possui maior largura de banda que a proposta neste projeto.

2.5 PLANEJAMENTO DE RISCOS

Projeto: Osciloscópio digital didático de baixo custo

1° Etapa: **Identificação do Risco**

Denominação do risco**: Escolha de tecnologia inadequada para atender os requisitos do cliente.**	N° Identificação 01
Descrição do risco: Caso ocorra a escolha de tecnologia inadequada para o projeto, haverá necessidade da compra de outra tecnologia e, portanto, perca de tempo devido à encomenda e estudo. Se as tecnologias forem dependentes entre si a escolha errada de uma delas pode afetar as outras.	

2° Etapa: **Avaliação do Risco**

Impacto: O **5 (alto)** O4(média/alto) O3(médio) O2(médio/baixo) O1(baixo) Explique: Devido ao tempo escasso do projeto a equipe pode ficar sem tempo para refazer todo o trabalho já feito com a tecnologia errada.
Probabilidade: O5 (alta) O4(média/alta) O **3(média)** O2(média/baixa) O1 (baixa) Explique: Projetos como de Oficinas de Integração II, deram alguma experiência para os integrantes da equipe na escolha de tecnologias, porém devido a complexidade deste projeto a probabilidade do risco de escolha de uma tecnologia errada passa de média/baixa para média.

3° Etapa: **Desenvolvimento da Resposta ao Risco**

Ações, Responsáveis e Datas de Conclusão	
Estratégias e Ações para eliminar ou reduzir este risco (minimizar impacto e/ou probabilidade): Prevenir: A equipe realizará um estudo extensivo das opções tecnológicas que podem ser usadas, produzirá um documento listando todas as opções e as comparando a fim de fazer a melhor escolha, visando atender a todos os requisitos com qualidade. Além disto, este documento passará pela aprovação dos professores, os quais possuem mais experiência e conhecimento do que os integrantes da equipe. Transferir: - - - Mitigar/Aceitar: Utilizar o plano "B" da tecnologia analisada.	
Impacto reavaliado: 4 (média/alto)	**Probabilidade reavaliada: 1 (baixo)**

Elaborado por: Geovane, Vinicius, Henrique, Tiago	Data: 25/08/2011	☐ Respostas incluídas na WBS/Cronograma	Registros adicionais: Verso ou Anexos

Formulário sugerido por Gasnier, 2000 Editora IMAN e alterado por Wille

Projeto: Osciloscópio digital didático de baixo custo

1° Etapa: **Identificação do Risco**

Denominação do risco: **Subestimação da complexidade, ou dificuldade na implementação de software ou hardware**	N° Identificação 02
Descrição do risco: Devido à inexperiência dos membros da equipe há o risco de que a complexidade de etapas do projeto sejam subestimadas resultando, assim, no atraso do desenvolvimento do projeto.	

2° Etapa: **Avaliação do Risco**

Impacto: O **5 (alto)** O4(média/alto) O3(médio) O2(médio/baixo) O1(baixo)
Explique: Como o tempo para o desenvolvimento do projeto é escasso atrasos no cronograma podem inviabilizar o projeto ou fazer com que as qualidades e\ou funcionalidades sejam diminuídas afim de que o projeto ainda possa ser concluído no tempo de 10 semanas.

Probabilidade: O **5 (alta)** O4(média/alta) O3(média) O2(média/baixa) O1 (baixa)
Explique: A equipe não possui conhecimento suficiente para obter uma boa estimativa da complexidade das tecnologias e suas integrações envolvidas no projeto. Com isso pode haver o risco de que alguma etapa demore mais que o planejado no cronograma.

3° Etapa: **Desenvolvimento da Resposta ao Risco**

Ações, Responsáveis e Datas de Conclusão

Estratégias e Ações para eliminar ou reduzir este risco (minimizar impacto e/ou probabilidade):
Prevenir: Verificar com outras pessoas (professores e alunos) que já trabalharam com as tecnologias envolvidas quanto tempo aproximadamente é necessário para a realização de sua implementação.

Transferir: - - -

Mitigar/Aceitar: A execução das etapas de desenvolvimento do projeto sempre será feita aos pares, desta forma se houver subestimação em alguma etapa do cronograma, a dupla realizará mais trabalho durante a semana. Para que as funcionalidades e\ou qualidades do projeto não sejam comprometidas o cronograma envolverá uma "folga" de duração de duas semanas, que poderá ser utilizada para a finalização de determinada implementação.

Impacto reavaliado: 2 (médio)	**Probabilidade reavaliada: 2 (média/baixa)**

Elaborado por: Geovane, Vinicius, Henrique, Tiago	Data: 25/08/2011	☐ Respostas incluídas na WBS/Cronograma	Registros adicionais: Verso ou Anexos

Formulário sugerido por Gasnier, 2000 Editora IMAN e alterado por Wille

Projeto: Osciloscópio digital didático de baixo custo

1° Etapa: **Identificação do Risco**

Denominação do risco: **Atraso na entrega de algum componente eletrônico**	N° Identificação 03
Descrição do risco: Caso necessário a importação de equipamentos, pode ocorrer que a entrega demore mais que o esperado.	

2° Etapa: **Avaliação do Risco**

Impacto: O **5 (alto)** O4(média/alto) O3(médio) O2(médio/baixo) O1(baixo)
Explique: Se ocorrer um atraso muito grande de algum componente eletrônico a equipe terá que tentar achar substituto ou realizar uma nova encomenda por outro distribuidor, comprometendo assim as funcionalidades e \ou qualidades do projeto ou até inviabilizando o projeto por falta de tempo.

Probabilidade: O5 (alta) O **4(média/alta)** O3(média) O2(média/baixa) O1 (baixa)
Explique: Para alcançar todos os requisitos e funcionalidades exigidas pelo cliente, se for necessário à equipe importará componentes eletrônicos. Pode ser que importemos produtos e, portanto, existe a possibilidade de que os componentes sejam retidos na Receita Federal atrasando sua entrega.

3° Etapa: **Desenvolvimento da Resposta ao Risco**

Ações, Responsáveis e Datas de Conclusão

Estratégias e Ações para eliminar ou reduzir este risco (minimizar impacto e/ou probabilidade):
Prevenir: Evitar a importação. Caso não haja no Brasil o produto, a equipe utilizará sites recomendados por professores e lojas que preveem a demora máxima de um mês na entrega, algumas lojas da China demoram mais que um mês, em média, para a entrega de uma encomenda, esse tipo de local será evitado.

Transferir: - - -

Mitigar/Aceitar: A equipe também pode usar componentes emprestados de professores ou alunos da UTFPR até a chegada da importação dos próprios componentes. O cronograma deve ser bem organizado de forma que tempos de espera por encomendas estejam previstos, diminuindo o impacto da espera pelos componentes.

Impacto reavaliado: 3 (médio)	**Probabilidade reavaliada: 2 (média)**

Elaborado por: Geovane, Vinicius, Henrique, Tiago	Data: 25/08/2011	☐ Respostas incluídas na WBS/Cronograma	Registros adicionais: Verso ou Anexos

Formulário sugerido por Gasnier, 2000 Editora IMAN e alterado por Wille

Projeto: Osciloscópio digital didático de baixo custo

1° Etapa: **Identificação do Risco**

Denominação do risco: **Demora na entrega da PCI**	N° Identificação 04
Descrição do risco: Como a confecção da placa de circuito impresso é dificultada pela falta de equipamentos, a equipe contratará terceiros para sua confecção, havendo a possibilidade de atraso na entrega da placa.	

2° Etapa: **Avaliação do Risco**

Impacto: O5 (alto) **O4(média/alto)** O3(médio) O2(médio/baixo) O1(baixo)
Explique: Se a PCI demorar a ser entregue a equipe não poderá realizar testes e nem saber se o produto final terá todas as funcionalidades e qualidades desejadas pelo cliente.

Probabilidade: O5 (alta) O4(média/alta) **O3(média)** O2(média/baixa) O1 (baixa)
Explique: Como a equipe comprará apenas três placas de circuito impresso poderá ocorrer que a empresa contratada priorize pedidos com maior numero de placas de circuito impresso, atrasando assim sua entrega. Caso a confecção da PCI seja feita na UTFPR, a probabilidade do risco é menor.

3° Etapa: **Desenvolvimento da Resposta ao Risco**

Ações, Responsáveis e Datas de Conclusão

Estratégias e Ações para eliminar ou reduzir este risco (minimizar impacto e/ou probabilidade):
Prevenir: Analisar o tempo máximo esperado para confecção de placas do local onde faremos o pedido, e adaptar o cronograma para esse tempo de espera.

Transferir: - - -

Mitigar/Aceitar: A equipe fará o roteamento da placa o mais rápido possível para que o pedido seja feito o quanto antes. O cronograma deve ser bem organizado de forma que o tempo de espera pela PCI esteja previsto, diminuindo o impacto de sua espera.

Impacto reavaliado: 2 (médio/baixo)	**Probabilidade reavaliada: 2 (média/baixa)**

Elaborado por: Geovane, Vinicius, Henrique, Tiago	Data: 25/08/2011	☐ Respostas incluídas na WBS/Cronograma	Registros adicionais: Verso ou Anexos

Formulário sugerido por Gasnier, 2000 Editora IMAN e alterado por Wille

Projeto: Osciloscópio digital didático de baixo custo

1° Etapa: **Identificação do Risco**

Denominação do risco**: Falha de comunicação entre a equipe**	N° Identificação 05
Descrição do risco: Devido à equipe não poder se encontrar diariamente a comunicação terá que ser feita basicamente através de e-mails. E pela comunicação ser feita dessa forma pode ocorrer falha na interpretação de informações, ocasionando desentendimento na equipe.	

2° Etapa: **Avaliação do Risco**

Impacto: ○5 (alto) **●4(média/alto)** ○3(médio) ○2(médio/baixo) ○1(baixo)

Explique: Se houver falha na comunicação entre a equipe o cronograma pode ser afetado, fazendo com que os membros da gastem mais tempo que 8 horas por dia para entregar o projeto no prazo.

Probabilidade: **●5 (alta)** ○4(média/alta) ○3(média) ○2(média/baixa) ○1 (baixa)

Explique: Os membros da equipe não se encontram pessoalmente todo dia. Portanto a probabilidade de utilizarmos outros recursos para a comunicação é alta.

3° Etapa: **Desenvolvimento da Resposta ao Risco**

Ações, Responsáveis e Datas de Conclusão

Estratégias e Ações para eliminar ou reduzir este risco (minimizar impacto e/ou probabilidade):

Prevenir: A equipe fará reuniões semanais (mais de uma, caso necessário) em que tudo deve ser esclarecido entre os membros. Cada membro fará um relatório semanal sobre o que fez e como fez, isso facilita os relatórios quinzenais para os clientes e também ajuda no entendimento da resolução do problema para os outros integrantes da equipe.

Transferir: - - -

Mitigar/Aceitar: Pedir esclarecimento absoluto para os membros de forma que não sejam deixadas dúvidas nas decisões de projeto. Fazer a releitura de materiais em texto produzidos pela equipe caso algo não seja esclarecido.

Impacto reavaliado: 3 (médio)	**Probabilidade reavaliada: 2 (média/baixa)**

Elaborado por: Geovane, Vinicius, Henrique, Tiago	Data: 25/08/2011	☐ Respostas incluídas na WBS/Cronograma	Registros adicionais: Verso ou Anexos

Formulário sugerido por Gasnier, 2000 Editora IMAN e alterado por Wille

Projeto: Osciloscópio digital didático de baixo custo

1° Etapa: **Identificação do Risco**

Denominação do risco**: Avaria em equipamentos com longo prazo de aquisição e de alto custo**	N° Identificação 06
Descrição do risco: Pode ocorrer de alguns equipamentos de alto custo que são necessários para o projeto estarem disponíveis somente via importação. Este risco trata de danos desses equipamentos e a necessidade de reimportar.	

2° Etapa: **Avaliação do Risco**

Impacto: O **5 (alto)** O4(média/alto) O3(médio) O2(médio/baixo) O1(baixo)
Explique: Danos em equipamentos com longo prazo de aquisição fazem com que seja necessária a importação de novos equipamentos e, portanto, o atraso do projeto.

Probabilidade: O **5 (alta)** O4(média/alta) O3(média) O2(média/baixa) O1 (baixa)
Explique: A probabilidade de ocorrer essa situação é alta, a cada semestre pelo menos uma equipe queima um microprocessador.

3° Etapa: **Desenvolvimento da Resposta ao Risco**

Ações, Responsáveis e Datas de Conclusão

Estratégias e Ações para eliminar ou reduzir este risco (minimizar impacto e/ou probabilidade):
Prevenir: As tarefas serão executadas em pares, uma pessoa montará o circuito e a outra será encarregada de verificar se todas as ligações foram feitas corretamente.

Transferir: - - -

Mitigar/Aceitar: Comprar mais de uma unidade de componentes via importação (três unidades).

Impacto reavaliado: 1 (baixo)	**Probabilidade reavaliada: 3 (média)**

Elaborado por: Geovane, Vinicius, Henrique, Tiago	Data: 25/08/2011	☐ Respostas incluídas na WBS/Cronograma	Registros adicionais: Verso ou Anexos

Formulário sugerido por Gasnier, 2000 Editora IMAN e alterado por Will

Projeto: Osciloscópio digital didático de baixo custo

1° Etapa: **Identificação do Risco**

Denominação do risco**: Falha na comunicação entre estação base e o microcontrolador.**	N° Identificação 07
Descrição do risco: A comunicação é um dos pré-requisitos do projeto na disciplina. Sua falha deve ser prevenida para o sucesso do projeto.	

2° Etapa: **Avaliação do Risco**

Impacto: O **5 (alto)** O4(média/alto) O3(médio) O2(médio/baixo) O1(baixo)
Explique: A falha de comunicação entre estação base e o microcontrolador inviabiliza o projeto.

Probabilidade: O5 (alta) O4(média/alta) O3(média) O **2(média/baixa)** O1 (baixa)
Explique: A comunicação entre estação-base e microcontrolador ocorrerá via USB, e há bastante documentação a respeito na internet e livros.

3° Etapa: **Desenvolvimento da Resposta ao Risco**

Ações, Responsáveis e Datas de Conclusão

Estratégias e Ações para eliminar ou reduzir este risco (minimizar impacto e/ou probabilidade):
Prevenir: No momento de analise de alternativas tecnológicas deverá ser verificado como é feita a comunicação do microcontrolador com outros dispositivos. Após a chegada do microcontrolador e da PCI, trabalhar na comunicação de dados entre os dispositivos como uma das primeiras etapas do cronograma, para verificar se ela realmente é possível.

Transferir: - - -

Mitigar/Aceitar: Importação de outra tecnologia.

Impacto reavaliado: 4(média/alto)	**Probabilidade reavaliada: 1 (baixa)**

Elaborado por: Geovane, Vinicius, Henrique, Tiago	Data: 25/08/2011	☐ Respostas incluídas na WBS/Cronograma	Registros adicionais: Verso ou Anexos

Formulário sugerido por Gasnier, 2000 Editora IMAN e alterado por Wille.

Projeto: Osciloscópio digital didático de baixo custo

1° Etapa: **Identificação do Risco**

Denominação do risco**: Não cumprimento dos prazos das metas estabelecidas pela equipe.**	N° Identificação 08
Descrição do risco: Metas serão estabelecidas para serem entregues a cada duas semanas (deliverables). Essas metas devem ser cumpridas a cada quinze dias e serão executadas aos pares.	

2° Etapa: **Avaliação do Risco**

Impacto: O **5 (alto)** O4(média/alto) O3(médio) O2(médio/baixo) O1(baixo) Explique: O impacto desse risco é alto porque atrasa outras metas (devido ao tempo que será necessário gastar para a meta não concluída, na semana seguinte ao deliverable), além de pressionar os integrantes da equipe psicologicamente.
Probabilidade: O5 (alta) O4(média/alta) O **3(média)** O2(média/baixa) O1 (baixa) Explique: No período de duas semanas cada membro da equipe tem 16 horas para o cumprimento da meta, nesse período professores ou outros alunos podem ser consultados, aconselhando a equipe para a resolução parcial dos objetivos.

3° Etapa: **Desenvolvimento da Resposta ao Risco**

Ações, Responsáveis e Datas de Conclusão	
Estratégias e Ações para eliminar ou reduzir este risco (minimizar impacto e/ou probabilidade): Prevenir: Planejar adequadamente o cronograma, colocando as metas principais para serem realizadas no início do projeto para que não atrasem futuramente outras metas dependentes delas. Transferir: - - - Mitigar/Aceitar: Utilizar o período de duas semanas de folga do cronograma	
Impacto reavaliado: 4 (médio/alto)	**Probabilidade reavaliada: 1 (baixa)**

Elaborado por: Geovane, Vinicius, Henrique, Tiago	Data: 25/08/2011	☐ Respostas incluídas na WBS/Cronograma	Registros adicionais: Verso ou Anexos

Formulário sugerido por Gasnier, 2000 Editora IMAN e alterado por Wille

Projeto: Osciloscópio digital didático de baixo custo

1° Etapa: **Identificação do Risco**

Denominação do risco**: Desistência de um membro da equipe.**	N° Identificação 09
Descrição do risco: Algum membro pode desistir da disciplina no meio do projeto.	

2° Etapa: **Avaliação do Risco**

Impacto: O5 (alto) **O4(média/alto)** O3(médio) O2(médio/baixo) O1(baixo)
Explique: A desistência de um membro da equipe afeta totalmente o cronograma e a distribuição de tarefas para cada integrante.

Probabilidade: **O5 (alta)** O4(média/alta) O3(média) O2(média/baixa) O1 (baixa)
Explique: Baixa probabilidade pela conversa que houve entre os integrantes e nenhum pretender desistir.

3° Etapa: **Desenvolvimento da Resposta ao Risco**

Ações, Responsáveis e Datas de Conclusão

Estratégias e Ações para eliminar ou reduzir este risco (minimizar impacto e/ou probabilidade):
Prevenir: Manter os membros animados com o projeto. Cabe ao gerente a tarefa de conversar semanalmente com os integrantes, para verificar o interesse e disponibilidade dos outros membros com o projeto.

Transferir: - - -

Mitigar/Aceitar: Planejamento do cronograma de forma que possam ser distribuídas as tarefas entre os integrantes, mesmo com a desistência de um integrante em qualquer momento do projeto, homogeneamente. Além disto, cada atividade será feita por dois membros da equipe, portanto nenhuma atividade dependerá só de um membro, e caso um integrante desista ainda haverá o integrante "backup".

Impacto reavaliado: 3 (médio)	**Probabilidade reavaliada: 2 (média)**

Elaborado por:	Data:	☐ Respostas incluídas na WBS/Cronograma	Registros adicionais: Verso ou Anexos
Geovane, Vinicius, Henrique, Tiago	25/08/2011		

Formulário sugerido por Gasnier, 2000 Editora IMAN e alterado por Wille

Projeto: Osciloscópio digital didático de baixo custo

1° Etapa: **Identificação do Risco**

Denominação do risco**: Erro de planejamento**	N° Identificação 10
Descrição do risco: Cronograma mal desenvolvido.	

2° Etapa: **Avaliação do Risco**

Impacto: O **5 (alto)** O4(média/alto) O3(médio) O2(médio/baixo) O1(baixo) Explique: Caso as funcionalidades ou a qualidade do artefato sejam afetados, os clientes avaliarão negativamente o produto, e pedirão que as etapas envolvidas sejam refeitas.
Probabilidade: O **5 (alta)** O **4(média/alta)** O3(média) O2(média/baixa) O1 (baixa) Explique: Devido à inexperiência da equipe em especificação do cronograma.

3° Etapa: **Desenvolvimento da Resposta ao Risco**

Ações, Responsáveis e Datas de Conclusão	
Estratégias e Ações para eliminar ou reduzir este risco (minimizar impacto e/ou probabilidade): Prevenir: Através de alunos ou professores experientes verificar a quantidade de tempo aproximada para a realização aproximada de cada tarefa. O cronograma deverá ser analisado por todos os membros da equipe, de forma que sejam planejados os períodos em que os membros não poderão utilizar horas extras (períodos de prova) para o desenvolvimento do projeto, transferindo essas horas para outras datas, como por exemplo, no início do projeto. Elaborar o cronograma de forma que haja um tempo de "folga" de duas semanas. Transferir: - - - Mitigar/Aceitar: Atualizar o cronograma caso as etapas não forem cumpridas. Fazer o uso das duas semanas de folga para a realização das tarefas mal planejadas.	
Impacto reavaliado: 3 (médio)	**Probabilidade reavaliada: 2 (média/baixa)**

Elaborado por: Geovane, Vinicius, Henrique, Tiago	Data: 25/08/2011	☐ Respostas incluídas na WBS/Cronograma	Registros adicionais: Verso ou Anexos

Formulário sugerido por Gasnier, 2000 Editora IMAN e alterado por Wille

2.6 ALTERNATIVA TECNOLÓGICA

Será apresentada neste tópico a análise realizada pela equipe em cima das diversas opções tecnológicas que envolvem este projeto. As decisões tomadas ao fim desta análise afetam diretamente a alocação de recursos e o cronograma que será seguido pela equipe. Levaremos em consideração os requisitos anteriormente descritos e o orçamento limite imposto pelo sponsor para definir a tecnologia mais apropriada.

2.6.1 HARDWARE

Segue abaixo a análise das opções tecnológicas referentes ao Hardware do projeto.

2.6.1.1 Microcontrolador

Os microcontroladores pesquisados deveriam atender a requisitos de velocidade, preço e também, possuir uma interface USB 2.0 Full Speed. Um ADC interno que satisfizesse os requisitos era desejável, porém não necessário visto que esse componente poderia ser comprado separadamente.

Os microcontroladores pesquisados deveriam atender a requisitos de velocidade, preço e também, possuir uma interface USB 2.0 Full Speed. Um ADC interno que satisfizesse os requisitos era desejável, porém não necessário visto que esse componente poderia ser comprado separadamente.

Optou-se por microcontroladores que possuíam uma versão da placa de desenvolvimento, o principal motivo disso é o alívio do risco de haver problemas com a PCI desenvolvida pela equipe.

Segue a tabela com os componentes que atendem os requisitos do projeto:

UC	STM32F103	*PIC32MX3	C8051T32
Velocidade de Processamento	72MHz(90DMIPS)	80MHz(80MIPS)	48MHz(48MIPS)
Número de ADC's	2	1	1
Número de bits ADC	12	10	10
Taxa de amostragem	1 MSPS	1MSPS	500KSPS

USB	2.0 Full Speed	2.0 OTG full speed	USB full speed 2.0
Debug	JTAG	JTAG	JTAG
Preço do uC	R$ 32,00	R$ 25,78	R$ 20,00
Preço da Placa de desenvolvimento	R$ 102,00	R$ 82,03	R$ 347,00
Preço do Programador	R$ 41,75	R$ 311,80	R$ 57,40
Disponibilidade no Brasil - uC	Sim	Sim	Não
Disponibilidade no Brasil - Placa des.	Não	Não	Não
Disponibilidade no Brasil - Programador	Não	Não	Não

Tabela 2 - Componentes que atendem os requisitos do projeto.

Importante enfatizar que o kit de desenvolvimento do PIC32 não possui pinos I/O, portanto é necessário comprar uma placa expansiva de $72.00.

Além destes microcontroladores foram verificadas outras empresas como *Texas Instruments*, *National Semicondutor*, *Analog Devices*, *Atmel* e *NXP*. Porém nenhum outro microcontrolador atendia todos os requisitos da equipe para o projeto (possuir microcontrolador, placa de desenvolvimento e programador). A Tabela **3** mostra outros micontroladores estudados que não atendem pelo menos uma necessidade do projeto.

uC	PIC24E	LM3S9B9	ATXmega
Velocidade de Processamento	60MHz(60MIPS)	80MHZ(100DMIPS)	32MHZ(32MIPS)
Número de ADC's	2	2	2
Número de bits ADC	10	10	10
Taxa de amostragem	1.1MSPS	1MSPS	1MSPS
USB	2.0 OTG full speed	2.0 OTG full speed	2.0 Full Speed
Debug	x	JTAG	JTAG e SWD
Preço do uC	x	x	x
Preço da Placa de desenvolvimento	R$ 106,60	R$ 162,70	x
Preço do Programador	x	x	x
Disponibilidade no Brasil - uC	x	x	x
Disponibilidade no Brasil - Placa de des.	Não	Não	x

Disponibilidade no Brasil - Programador	x	x	x

Tabela 3 - Componentes que não atendem os requisitos do projeto.

PIC24E também não possui pinos I/O, portanto sendo necessária a compra de uma placa expansiva de $72.00.

Alternativa escolhida:

O microcontrolador escolhido foi o STM32F103. Os principais motivos são: quantidade suficiente para venda na farnell (mais de 50 unidades em estoque local), em comparação com o PIC32 que possui somente 2 unidades em estoque local e o C8051T32 que só há disponível via importação. O preço dos três itens (microcontrolador, placa de desenvolvimento e programador) para o caso do PIC e do C8051 aumentaria o custo de projeto podendo ultrapassar a cota de 700 reais. O STM32 também possui o ADC embutido no microcontrolador, portanto, diminuindo o custo da placa de circuito impresso que será feita pela equipe.

2.6.1.2 ADC

A Tabela **4** com os componentes que atendem os requisitos do projeto, o principal objetivo da realização dessa pesquisa, foi que o PIC32 não possuía 2 ADCs embutidos, sendo necessário a utilização de pelo menos mais um ADC:

ADC	Num. ADC's embutidos	Num de Canais	Num. bits ADC	Taxa de amostragem	Interface	Preço	Disponibilidade no Brasil
MAX11102	1	2	12	3MSPS	Serial	R$7,7	Não
AD7352	2	2	12	3MSPS	Serial	R$21,3	Não

Tabela 4 - ADCs Pesquisados

Alternativa escolhida:

Nenhuma, em decorrência do processador escolhido já possuir 2 ADC's que funcionam de acordo com a necessidade do projeto.

2.6.1.3 Charge Pump (Fonte Chaveada)

A alimentação do USB provê 5V que serão utilizados para a alimentação tanto do microcontrolador quanto para alimentação positiva dos amplificadores, porém para a alimentação negativa dos amplificadores utilizaremos uma fonte chaveada que inverte o sinal de alimentação (USB) para -5V (o valor da saída dependerá da eficiência da fonte chaveada utilizada).
Algumas opções pesquisadas:

Charge Pump	Eficiência na Conversão	Preço	Disponibilidade no Brasil
TC1221	96.0%	-	-
LM2662	>90%	R$ 5.6	Sim
TPS6040	>90%	R$ 2.3	Sim

Tabela 5 - Opções pesquisadas de fonte chaveada

Alternativa escolhida:

A opção escolhida foi o TPS6040, pois está disponível na Farnell e possui oscilador interno. O LM2662 necessita de um oscilador para funcionamento, sendo mais um componente extra na PCI.

2.6.1.4 Amplificadores

Os amplificadores que a equipe utilizará terão sinal de entrada de até 100kHz. Opções acessíveis e existentes no mercado são:

Amp.	Numero de amp. no CI	Encap.	Faixa de frequência máxima com um ganho unitário (MHz)	CMRR (DB)	Slew Rate(V/us)	Ruído nV/H^(1/2)	Impedância de entrada	Preço
TL082	2	DIP	3.0	86	13	18	1000 MΩ	R$ 1,97
LM318	1	Mini-DIP	15	100	50	-	3 MΩ	R$ 2,12

Tabela 6 - Amplificadores pesquisados

Alternativa escolhida:

O CI para amplificador operacional escolhido foi o TL082, pois ele possui alta impedância de entrada por ser um amplificador operacional BI-FET e as outras especificações são suficientes para o projeto de aquisição de sinal

proposto. O segundo motivo da escolha deste CI é por possuir dois amplificadores operacionais embutidos e, portanto, ocupando menos espaço na placa de circuito impresso que será feita pela equipe.

2.6.2 SOFTWARE

2.6.2.1 Linguagem de Programação de Estação Base

Foram utilizadas na escolha da linguagem de programação critérios como: prover biblioteca de interface com USB e API gráfica. As opções encontradas foram Java, C++. Segue uma tabela com pontos importantes elencados pela equipe.

Linguagem	Possui API para Interface USB 2.0	A API possui versão gratuita	Obriga ao uso de um sistema operacional	API gráfica nativa
Java	Sim, possui a JSR80 e jUSB	Sim, ambas as APIs são de graça e open source	A API jUSB só pode ser usada com GNU/Linux, e a API JSR80 possui uma versão alpha para Windows	sim
C++	Sim, o nome é USBIO	Sim, porém a versão gratuita funciona apenas durante 4 horas, após isso o computador tem que ser reiniciado.	Sim, só pode ser usado com várias versões do Windows desde o 98/ME ate o Windows 7, além disso suporta Windows 32 e 64 bits	não

Tabela 7 - Comparações linguagens de programação

Alternativa escolhida:

A linguagem de programação escolhida para o projeto foi Java. A linguagem em C++ também atende os requisitos do projeto, porém a escolha entre as duas opções foi feita baseada na experiência prévia da equipe com projetos que possuem interface gráfica.

2.6.2.2 Sistema Operacional da Estação Base

Dentre os sistemas operacionais disponíveis no mercado podemos citar os seguintes:

Windows, Linux e MacOS. O presente projeto visa utilizar o sistema que obtém maior acesso e facilidade por parte dos sujeitos que irão fazer o uso do osciloscópio.

Alternativa escolhida:

Devido à popularidade que o Windows exerce sobre os demais sistemas operacionais, a equipe optou por utilizá-lo, percebemos que entre os alunos de eletrônica e computação da UTFPR a maioria tem como Windows seu Sistema Operacional principal.

Não faremos o programa multiplataforma, pois a biblioteca USB da linguagem escolhida (Java) para Linux é diferente da biblioteca para Windows (essas bibliotecas não são nativas). Portanto não podendo ser utilizado a portabilidade Java para ambos os Sistemas Operacionais e seria inviável para o projeto produzir versões para as duas plataformas por restrições severas de tempo em que o projeto se encontra.

2.6.2.3 Ambiente integrado para o desenvolvimento do software da Estação Base

O software a ser implementado utilizará um ambiente de desenvolvimento integrado (IDE), que se divide principalmente em Eclipse e NetBeans, estes são frequentemente vistos em Java, podendo também dar suporte para outras linguagens.

Alternativa escolhida:

Diante das opções de ambientes disponíveis, sejam estas Eclipse e NetBeans, a equipe tem maior familiaridade por utilizar a segunda opção. Cumpre esclarecer que não há grande diferença técnica entre os ambientes, sendo assim a qualidade técnica não exerce nenhuma influencia no desenvolvimento do projeto.

2.6.2.4 Programador e Debugger

O programador/debugger escolhido pela equipe para o microcontrolador STM32F103 foi o ST-Link/V2, o principal motivo da escolha desse programador foram: disponibilidade no mercado (mouser.com). Outros programadores/debuggers como o STX-RLink não foram escolhidos, pois a IDE que dá suporte a este programador é pago e não possui versão de demonstração.

2.6.2.5 Ambiente integrado para o desenvolvimento do software do microcontrolador.

A IDE escolhida pela equipe para o microcontrolador será o Keil uVision 4.2, pois é compatível com o programador e possui uma versão demo. A versão demo possui limitações que não afetam o projeto.

Esta IDE foi produzida pelo fabricante do core do microcontrolador que utilizaremos (ARM), e o debugger atende os requisitos do projeto, por exemplo, é possível analisar a saída do ADC interno ao microcontrolador em uma janela gráfica.

2.6.2.6 Linguagem de Programação do microcontrolador.

Para as configurações de registradores e rotinas localizadas em regiões críticas, que precisaremos de processamento eficiente, utilizaremos assembly. Em outras partes do código será utilizado rotinas em C, para chamada de funções.

2.7 ORÇAMENTO INICIAL DETALHADO PREVISTO

Este tópico apresenta uma estimativa de custo total do projeto. Como ele se encontrava numa fase inicial, a estimativa girou em torno de várias opções de soluções para atenderem aos problemas propostos, não apresentando necessariamente o preço realmente gasto na execução muito menos os produtos que serão efetivamente utilizados. Em suma, é apresentada uma estimativa de custo total do projeto:

Para componentes estimamos a necessidade de:

- Microcontrolador com:
  - Interface USB
  - 2 canais ADC 12bits com taxa de conversão acima de 1 Msps com conversão simultânea.
  - Capaz de transportar os dados do ADC para o buffer USB em uma taxa de 2 MB/s.
- Amplificador Operacional.
- Fonte chaveada de 5V (USB) para -5V.
- Placa de circuito impresso.
- Placa de desenvolvimento
- Componentes eletrônicos.
- Conectores e cabo USB.

Meta de gastos para componentes: R$ 710,00

Além dos custos materiais, existem os custos humanos. Em uma equipe de 4 elementos cada membro precisa trabalhar em média 8 horas por semana durante as 10 semanas de desenvolvimento do projeto. Estimando a hora trabalhada em R$ 40,00, o custo de recursos humanos ao final do projeto é de aproximadamente R$ 12.800,00.

2.8 ORÇAMENTO DETALHADO

Componentes	Preço	Quantidade	Orçamento final
uC	R$ 32,00	3	R$ 96,00
kit de desenvolovimento	R$ 102,00	1	R$ 102,00
programador JTAG	R$ 41,75	1	R$ 41,75
PCI (3 unidades)	R$ 200,00	1	R$ 200,00
Componentes	R$ 50,00	1	R$ 50,00
Frete	R$ 150,00	1	R$ 150,00
Amplificador TL082	R$ 2,00	5	R$ 10,00
Fonte Chaveada	R$ 2,30	3	R$ 6,90
Cabo USB	R$ 5,00	1	R$ 5,00
Multiplexador	R$ 15,00	2	R$ 30,00
Total	R$ 600,05		R$ 691,65

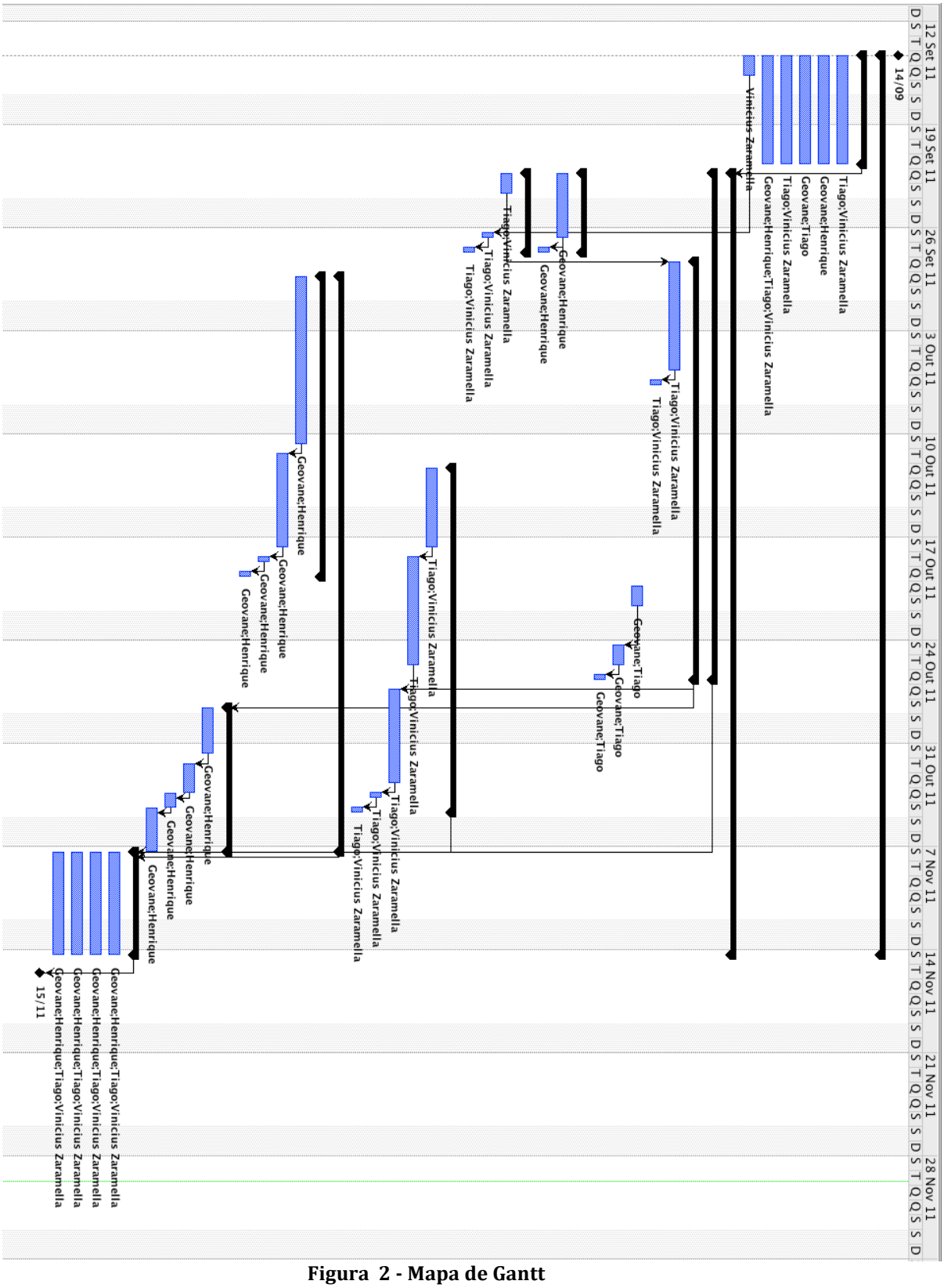
Tabela 8 - Orçamento detalhado previsto

2.9 CRONOGRAMA

Para o desenvolvido do cronograma utilizou-se o software livre OpenProj. A Figura 1 mostra o cronograma seguido no desenvolvimento do projeto pela nossa equipe.

		Nome	Duração	Início	Término	Prede...	Nome do Recurso
1		Marco de Início	0 dias	14/09/11 08:00	14/09/11 08...		
2		**Execução**	43,25 dias?	**14/09/11 08:00**	**14/11/11 1...**		
3		**Estudo e Relátorio Técnico**	6 dias	**14/09/11 08:00**	**21/09/11 1...**		
4		Sistema de Aquisição de Dados	6 dias	14/09/11 08:00	21/09/11 17...		Tiago;Vinicius Zaramella
5		Microcontrolador	6 dias	14/09/11 08:00	21/09/11 17...		Geovane;Henrique
6		Comunicação USB	6 dias	14/09/11 08:00	21/09/11 17...		Geovane;Tiago
7		Modelagem de PCI	6 dias	14/09/11 08:00	21/09/11 17...		Tiago;Vinicius Zaramella
8		Funcionamento Ociloscópio	6 dias	14/09/11 08:00	21/09/11 17...		Geovane;Henrique;Tiago;Vinicius Zaramella
9		Compra de Componentes	2 dias	14/09/11 08:00	15/09/11 17...		Vinicius Zaramella
10		**Desenvolvimento**	37,25 dias?	**22/09/11 08:00**	**14/11/11 1...**	**3**	
11		**Hardware**	25 dias	**22/09/11 08:00**	**26/10/11 1...**		
12		**PCI**	21 dias	**28/09/11 08:00**	**26/10/11 1...**		
13		Projeto	6 dias	28/09/11 08:00	05/10/11 17...	22;19	Tiago;Vinicius Zaramella
14		Pedido	1 dia	06/10/11 08:00	06/10/11 17...	13	Tiago;Vinicius Zaramella
15		Montagem	2 dias	20/10/11 08:00	21/10/11 17...		Geovane;Tiago
16		Teste	2 dias	24/10/11 08:00	25/10/11 17...	15	Geovane;Tiago
17		Documentação	1 dia	26/10/11 08:00	26/10/11 17...	16	Geovane;Tiago
18		**Microcontrolador**	4 dias	**22/09/11 08:00**	**27/09/11 1...**		
19		Projeto do Circuito	3 dias	22/09/11 08:00	26/09/11 17...		Geovane;Henrique
20		Documentação	1 dia	27/09/11 08:00	27/09/11 17...	19	Geovane;Henrique
21		**Sistema de Aquisição de Dados**	4 dias	**22/09/11 08:00**	**27/09/11 1...**		
22		Projeto	2 dias	22/09/11 08:00	23/09/11 17...		Tiago;Vinicius Zaramella
23		Teste	1 dia	26/09/11 08:00	26/09/11 17...	9;22	Tiago;Vinicius Zaramella
24		Documentação	1 dia	27/09/11 08:00	27/09/11 17...	23	Tiago;Vinicius Zaramella
25		**Comunicação**	18 dias	**12/10/11 08:00**	**04/11/11 1...**		
26		Estudo de Protocolos	4 dias	12/10/11 08:00	17/10/11 17...		Tiago;Vinicius Zaramella
27		Projeto do Protocolo	6 dias	18/10/11 08:00	25/10/11 17...	26	Tiago;Vinicius Zaramella
28		Implementação	5 dias	27/10/11 08:00	02/11/11 17...	12;27	Tiago;Vinicius Zaramella
29		Teste	1 dia	03/11/11 08:00	03/11/11 17...	28	Tiago;Vinicius Zaramella
30		Documentação	1 dia	04/11/11 08:00	04/11/11 17...	29	Tiago;Vinicius Zaramella
31		**Software**	27,25 dias?	**29/09/11 08:00**	**07/11/11 1...**		
32		**Estação Base**	15 dias?	**29/09/11 08:00**	**19/10/11 1...**		
33		Projeto	8 dias?	29/09/11 08:00	10/10/11 17...		Geovane;Henrique
34		Implementação	5 dias	11/10/11 08:00	17/10/11 17...	33	Geovane;Henrique
35		Teste	1 dia	18/10/11 08:00	18/10/11 17...	34	Geovane;Henrique
36		Documentação	1 dia	19/10/11 08:00	19/10/11 17...	35	Geovane;Henrique
37		**Sistema Embarcado**	5,625 dias	**28/10/11 14:00**	**07/11/11 1...**	**12**	
38		Projeto	1,375 dias	28/10/11 14:00	31/10/11 17...		Geovane;Henrique
39		Implementação	2 dias	01/11/11 10:00	03/11/11 10...	38	Geovane;Henrique
40		Teste	1 dia	03/11/11 10:00	04/11/11 10...	39	Geovane;Henrique
41		Documentação	1 dia	04/11/11 10:00	07/11/11 10...	40	Geovane;Henrique
42		**Sistema Final**	5 dias	**07/11/11 10:00**	**14/11/11 1...**	**11;2...**	
43		Testes do Sistema Final	5 dias	07/11/11 10:00	14/11/11 10...		Geovane;Henrique;Tiago;Vinicius Zaramella
44		Documentação de Testes	5 dias	07/11/11 10:00	14/11/11 10...		Geovane;Henrique;Tiago;Vinicius Zaramella
45		Relátorio Final	5 dias	07/11/11 10:00	14/11/11 10...		Geovane;Henrique;Tiago;Vinicius Zaramella
46		Criar Apresentação	5 dias	07/11/11 10:00	14/11/11 10...		Geovane;Henrique;Tiago;Vinicius Zaramella
47		Marco de Fim	0 dias	15/11/11 15:00	15/11/11 15...	42	

Figura 1 – Cronograma


12 Set 11
19 Set 11
26 Set 11
3 Out 11
10 Out 11
17 Out 11
24 Out 11
31 Out 11
7 Nov 11
14 Nov 11
21 Nov 11
28 Nov 11
14/09
15/11
Tiago;Vinicius Zaramella
Geovane;Henrique
Geovane;Tiago
Tiago;Vinicius Zaramella
Geovane;Henrique;Tiago;Vinicius Zaramella
Geovane;Henrique;Tiago;Vinicius Zaramella
Geovane;Henrique;Tiago;Vinicius Zaramella
Geovane;Henrique;Tiago;Vinicius Zaramella
Geovane;Henrique;Tiago;Vinicius Zaramella
Geovane;Tiago
Geovane;Tiago
Geovane;Henrique
Geovane;Henrique
Geovane;Henrique
Geovane;Henrique
Geovane;Henrique
Geovane;Henrique
Geovane;Henrique
Geovane;Henrique
Geovane;Henrique
Geovane;Henrique
Tiago;Vinicius Zaramella
Tiago;Vinicius Zaramella
Tiago;Vinicius Zaramella

2.10 ENTREGÁVEIS

Dia	Auxiliar de gerenciamento	Deliverable
28/09	Tiago Rafael Volkmann Coelho	-Testes com o circuito de aquisição de dados: circuito de entrada com amplificador programável em protoboard. - Projeto do circuito esquemático do microcontrolador .
19/10	Henrique Reinaldo Sarmento	- Projeto de software da estação base. Casos de uso e diagramas de classe. - Projeto da placa de circuito impresso do microcontrolador e sistema de aquisição de dados.
9/11	Henrique Reinaldo Sarmento	- Hardware do protótipo pronto, com a PCI e tudo soldado nela. Demonstração da aquisição de sinais pelo debugger. - Projeto do protocolo de comunicação.
23/11	Geovane Vinicius Ferreira	- Comunicação entre microcontrolador e estação base funcionando com o protocolo de comunicação. - Projeto do software do sistema embarcado - Software da estação base e do microcontrolador funcionando. Demonstração do funcionamento.

Tabela 9 - Deliverables e auxiliar de gerente

3 FUNDAMENTAÇÃO TEÓRICA

3.1 OSCILOSCÓPIO

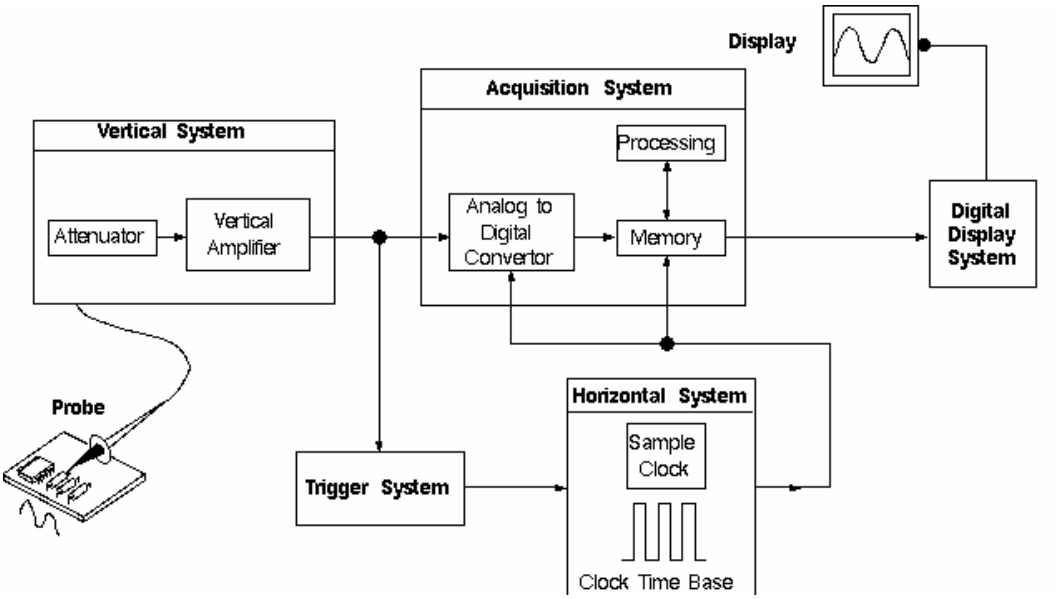


Figura 3 - Diagrama de blocos osciloscópio

Neste tópico faremos uma breve revisão das características pertinentes ao osciloscópio digital, no diagrama da Figura 3 temos um diagrama de blocos exemplificando o sistema de um osciloscópio. O sistema vertical é descrito nesse relatório como o condicionamento de sinais, e o sistema de aquisição de dados que é descrito neste tópico pelo ADC.

3.1.1 Funcionalidades

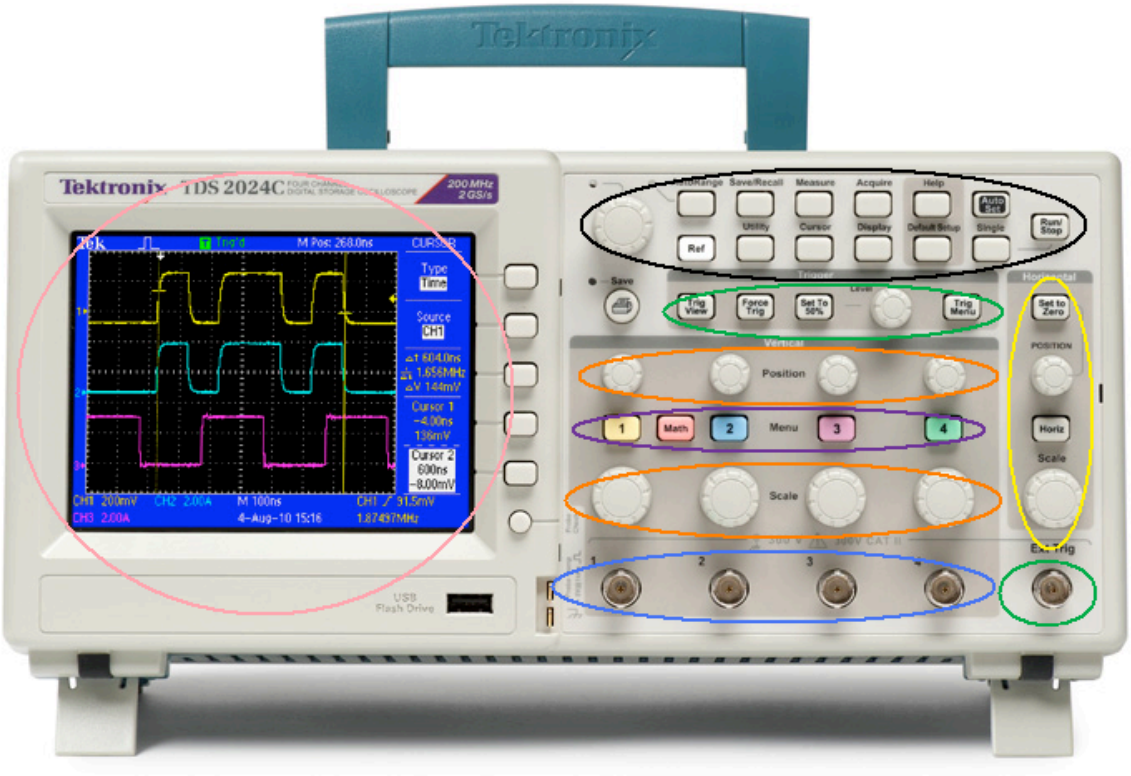


Figura 4 - Painel frontal de um osciloscópio digital

Os principais botões de um osciloscópio estão mostrados na Figura 4, neste tópico será descrito algumas funcionalidades que esses botões possuem:

- Como podemos observar na imagem acima o circulo em rosa mostra o gráfico (plotador) onde os sinais podem ser analisados, nesse osciloscópio temos a possibilidade de verificar sinais de até 4 canais.
- O circulo em azul contém a entrada dos canais onde o usuário encaixa a ponta de prova e pode medir o sinal que deseja.
- No circulo **preto** está contido os menus gerais do osciloscópio, além disso, botões com funcionalidades de RUN/STOP que congela/descongela o gráfico que está sendo plotado, *measure* que o usuário pode verificar medições de tensão pico-a-pico, RMS, média, máxima, mínima e a freqüência do sinal no menu *acquire* pode ser selecionado o tipo de aquisição em que o osciloscópio irá fazer a plotagem dos sinal adquirido, e por fim há nesse conjunto de botões a seleção de opções envolvendo

cursores que auxiliam na medição de período e tensão dos sinais.

- No circulo verde temos a função de *trigger*, a principal função do *trigger* é fazer com que o sinal medido fique estabilizado e não "correndo" na tela.
- No circulo roxo há botões que selecionam opções dos canais como acoplamento AC ou DC, além disso, o botão Math que pode ser analisado por exemplo a FFT do sinal.
- O circulo laranja contém operações com escala de tensão e a posição vertical do sinal (*offset*).
- O circulo amarelo contém operações com escala da base de tempo e a posição horizontal do sinal (*offset*).

3.1.2 Condicionamento de Sinais

A função do condicionamento de sinal nos osciloscópios é fazer com que a tensão na entrada torne-se adequada para o ADC que possui seus limites de entrada e referência, no projeto de [1] o condicionamento de sinal foi feito utilizando divisores resistivos, para a atenuação, e amplificadores operacionais que amplificam o sinal e fazem seu deslocamento. Também percebemos o uso de capacitores no circuito de condicionamento de sinal em [1], a necessidade da utilização desses componentes é utilizada para a compensação de capacitâncias intrínsecas do circuito. Em [4] são feitos os cálculos que deduzem essa compensação.

3.1.3 Amostragem de Sinais Variantes no Tempo

3.1.3.1 Limite de Nyquist

O limite de Nyquist afirma que para a reconstrução de um sinal amostrado, é necessário no mínimo uma frequência de amostragem duas vezes maior que a maior componente de frequencial do próprio sinal. Apesar disso em osciloscópio são utilizadas frequências de amostragem de

cinco até dez vezes superiores que a maior componentes frequencial [6], o motivo de utilizar uma amostragem superior ao limite de Nyquist é para obter amostras representativas do sinal que está sendo amostrado e eliminar alising.

3.1.3.2 *Conversão Analógica Digital*

O conversor analógico-digital (ADC) é um dispositivo eletrônico capaz de gerar uma representação digital a partir de uma grandeza analógica, normalmente um sinal representado por um nível de tensão ou intensidade de corrente elétrica.

Os ADCs são muito úteis na interface entre dispositivos digitais (microprocessadores, microcontroladores, DSPs, etc) e dispositivos analógicos e são utilizados em aplicações como leitura de sensores, digitalização de áudio e vídeo.

Por exemplo, um conversor A/D de 10 bits, preparado para um sinal de entrada analógica de tensão variável de 0V a 5V pode assumir os valores binários de 0 (0000000000) a 1023 (1111111111), ou seja, é capaz de capturar 1024 níveis discretos de um determinado sinal. Se o sinal de entrada do suposto conversor A/D estiver em 2,5V, por exemplo, o valor binário gerado será 512.

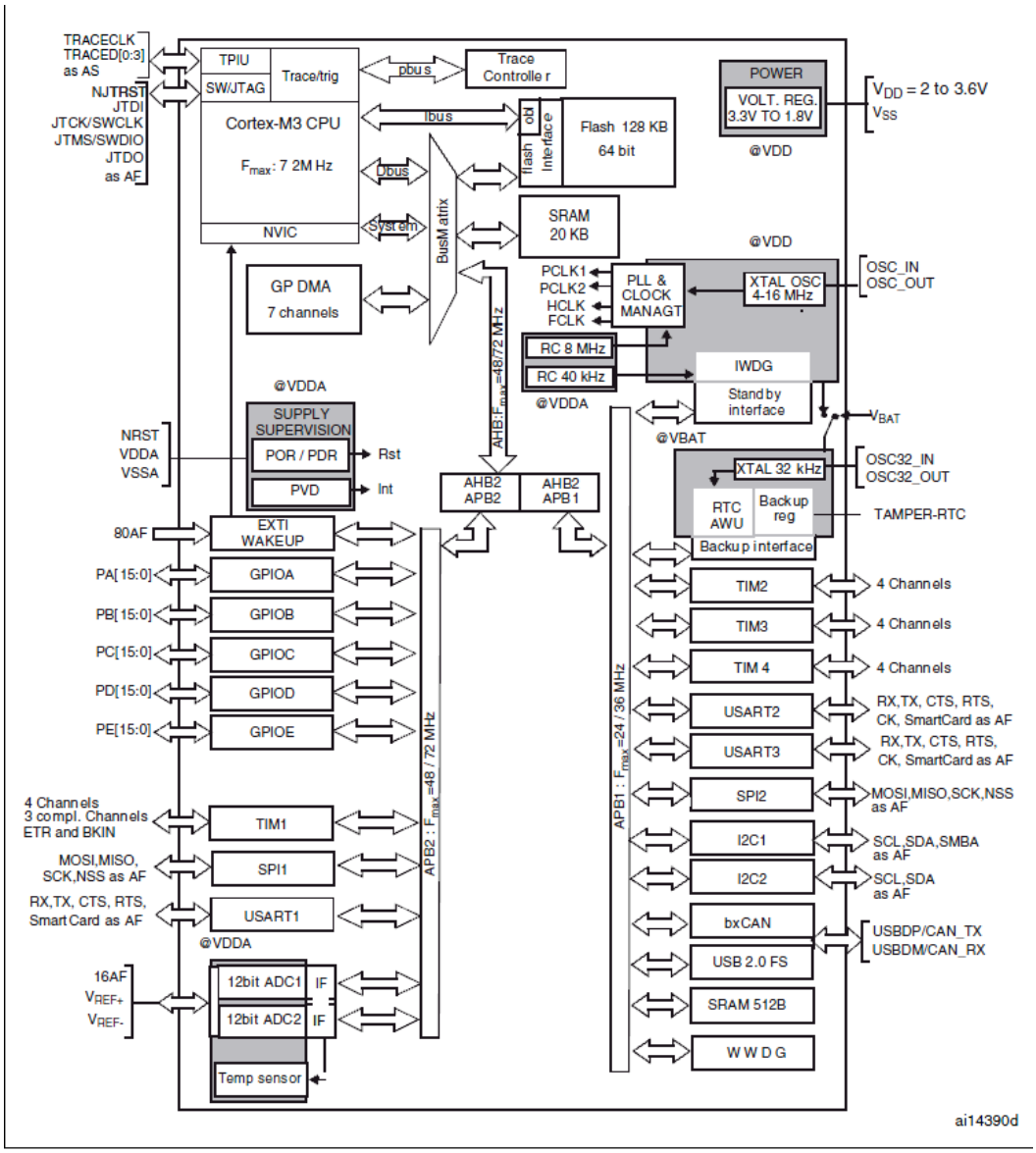
3.1.3.3 *Conversão A/D por Aproximação Sucessivas*

A aproximação sucessiva é um método de conversão de sinal analógico digital o qual possui tempo de amostragem constante, proporcional ao número de bits do conversor e ao sinal de clock.

3.2 STM32F103

A família STM32 de microcontroladores da empresa STMicroeletronics são microcontroladores de 32-bits baseados na arquitetura de processadores ARM CortexTM-M. Este oferece um produto de alta performance capaz de trabalhar com sistemas de tempo real e processamento digitais de sinais, também possui periféricos como ADC,

DAC, interface USB, DMA, Timer, entre outros. A Figura 5 mostra o diagrama de blocos do sistema deste microcontrolador.



1. T_A = –40 °C to +105 °C (junction temperature up to 125 °C).
2. AF = alternate function on I/O port pin.

Figura 5 - Diagrama de blocos do STM32

3.2.1 Core

Arquitetura ARM (Advanced RISC Machine) é uma arquitetura de processador de 32 bits que é usada principalmente em sistemas embarcados. Muito usada na indústria e na informática, seu desenvolvimento se deu visando obter o melhor desempenho possível, com

a limitação de ser simples, ocupar pouca área e ter baixo consumo de energia.

Possuem poucas instruções para programação. São encontrados em PDAs, telefones celulares, calculadoras, periféricos de computador, equipamentos POS e aplicações industriais.

Os processadores ARM são 90% dos processadores embarcados RISC de 32 bits.

Principais Características:

- Arquitetura Load-Store: as instruções somente processarão (soma, subtração, etc) valores que estiverem nos registradores e sempre armazenarão os resultados em algum registrador.
- Instruções fixas de 32 bits de largura (com exceção das instruções Thumb compactas de 16 bits) alinhadas em 4 bytes consecutivos da memória, com execução condicional, com poderosas instruções de carga e armazenamento de múltiplos registradores, capacidade de executar operações de deslocamento e na ULA com uma única instrução executada em um ciclo de clock .
- Formato de instruções de 3 endereços (isto é, os dois registradores operandos e o registrador de resultado são independentemente especificados)
- 15 registradores de 32 bits para uso geral
- Manipulação de periféricos de I/O como dispositivos mapeados na memória com suporte à interrupções.
- Conjunto de instruções aberto a extensões através de co-processador, incluindo a adição de novos registradores e tipos de dados ao modelo do programador.
- Pipelines de 3 e 5 estágios
- Baixo Consumo de energia;
- Tamanho do núcleo reduzido;

3.2.2 Periféricos

Os periféricos do microcontrolador STM32F103 utilizados neste projeto são os seguintes: ADC, DMA, USB, JTAG. Estes serão explicados a seguir.

3.2.2.1 ADC

Para mais informações sobre o funcionamento de conversores analógicos digitais vide 3.1.3.2.

O ADC existente do no microcontrolador STM32F103 tem as seguintes especificações:

2 Conversores de 12 bis

Tempo de conversão máximo de 1us

Faixa de conversão: 0 à 3,6 V

16 canais de conversão

Sensor de temperatura acoplado

3.2.2.2 Timers

Um temporizador (Timer) é um dispositivo capaz de medir o tempo, sendo um tipo de relógio especializado. Ele pode ser usado para controlar a sequência de um evento ou processo. Temporizadores podem ser mecânicos, eletromecânicos, digitais, ou mesmo programas de computador, uma vez que os computadores contêm relógios.

Na arquitetura de circuitos microprocessados, os temporizadores são utilizados para gerar bases de tempo que podem ser utilizadas para os mais diversos fins, como por exemplo gerar sinais de clock para outros periféricos do chip, calcular intervalos de tempo ou medir período de sinais.

Na arquitetura dos microcontroladores mais utilizadas atualmente no meio acadêmico "MCS51" e o "PIC16" estão presentes temporizadores de 8 e 16 bits, esses podendo ser configurados pelo usuário para trabalhar de formas distintas.

As características dos Timers existentes no microcontrolador STM32F103 são:

- 7 timers
- 3 x 16-bit timers, cada um com mais de 4 IC/OC/PWM ou contador de pulsos e entrada de encoder de quadrature (incremental).
- 16-bit, PWM para controle de motores com gerador de deadtime e parade de emergencia.
- 2 watchdog timers.
- SysTick timer 24-bit downcounter

3.2.2.3 DMA

O termo DMA é um acrónimo para a expressão em inglês Direct memory access. O DMA permite que certos dispositivos de hardware num computador acessem a memória do sistema para leitura e escrita independentemente da CPU. Muitos sistemas utilizam DMA, incluindo controladores de disco, placas gráficas, de rede ou de som.

O acesso direto da memória é usado igualmente para transferência de dados de núcleos em processadores multi-core, em especial nos sistema-em-microplaquetas do processador, onde seu elemento de processamento é equipado com uma memória local, e o acesso direto da memória é usado para transferir dados entre a memória local e a memória principal.

Os computadores que têm os canais de acesso direto a memória podem transferir dados aos dispositivos com muito menos perdas gerais de processamento do que computadores sem uma via de acesso direto à memória. Similarmente um elemento de processamento dentro de um

processador multi-core pode transferir dados para e de sua memória local sem ocupar seu tempo de processamento e permitir a simultaneidade de transferência de dados.

Sem acesso direto da memória, usando a modalidade programada de entrada/saída (E/S) para uma comunicação com os dispositivos periféricos, ou as instruções da troca no caso dos núcleos multi-core, o processador central é ocupado inteiramente para a leitura ou escrita da operação, e assim não se torna possível executar o outro afazer. Com acesso direto da memória, o processador central executa transferências, faz outras operações enquanto alguma transferência estiver em andamento, recebe uma interrupção do controlador de acesso direto da memória uma vez que a operação foi feita.

As características do DMA existente do microcontrolador STM32F103 são:

- 7 canais de DMA
- Suporta interação com Timers
- Suporta interação com ADC
- Suporta interação com SPI
- Suporta interação com I^2C
- Suporta interação com USARTs

3.2.2.4 JTAG

Joint Test Action Group (JTAG) é um nome comum para o que foi posteriormente oficializado como IEEE 1149.1 Standard Test Access Port and Boundaru-Scan Architecture. Este foi inicialmente utilizado para teste em placas de circuito impresso usando boundary scan e ainda é muito utilizado para este propósito.

JTAG é também muito utilizado para debug de portas de circuitos integrados. No mercado de processadores embarcados, essencialmente todos os processadores modernos suportam JTAG quando este tem pinos suficientes. Em desenvolvimento de sistemas embarcados este é utilizado para tarefas de debug como single stepping and breakpointing.

3.2.3 Encapsulamento

O microcontrolador STM32 possui algumas opção de encapsulamento. A opção escolhida para este projeto foi a LQFP-64 mostrada na Figura 6.

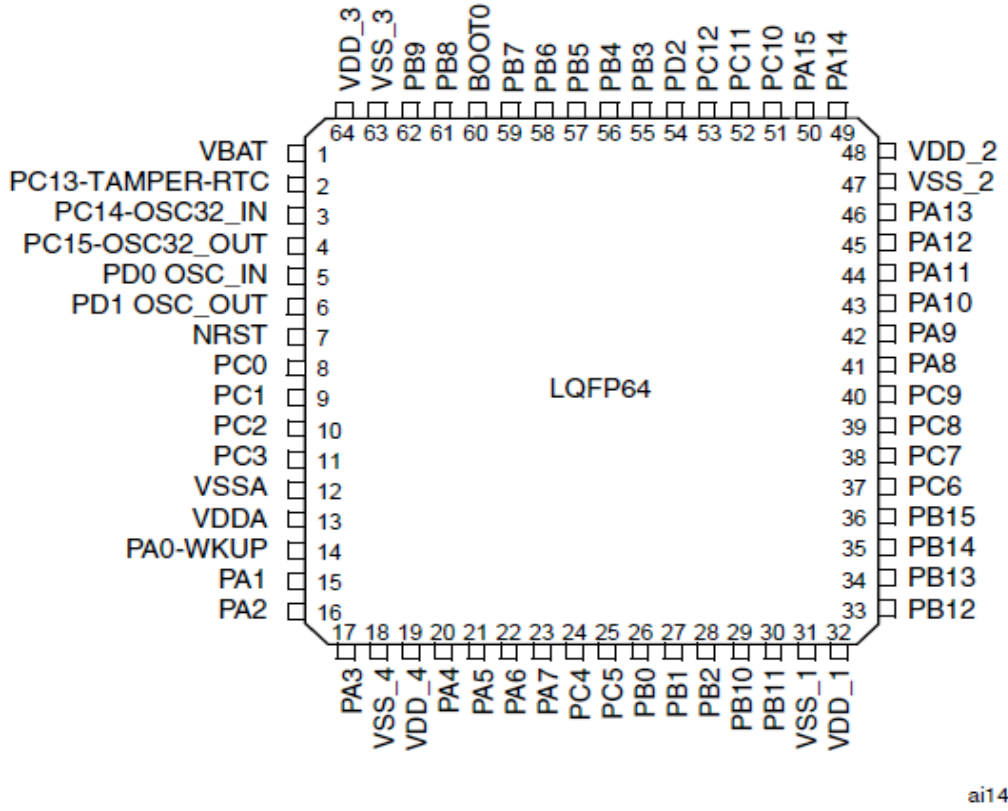


Figura 6 - Encapsulamente LQFP-64 STM32F103

3.2.4 Kit de Desenvolvimento

O kit de desenvolvimento utilizado neste projeto foi o STM-P103 da Olimex[10]. Este possui as seguinte características:

- CPU: STM32F103RBT6 ARM 32 bit CORTEX M3™
- Conector JTAG com pinagem ARM 2x10 para programação/*debugging.*
- Conector USB
- Driver e conector CAN
- Driver e conector RS232
- Conector UEXT
- Conector SD-MMC
- Conector para bateria de *backup*
- Botão de reset
- LED de status
- LED de alimentação
- Regulador de tensão de 3.3 V com mais de 800 mA de corrente
- Oscilador de cristal de 8 Mhz
- Oscilador de cristal de 32768 Hz para RTC

A Figura 7 mostra uma imagem do kit de desenvolvimento.

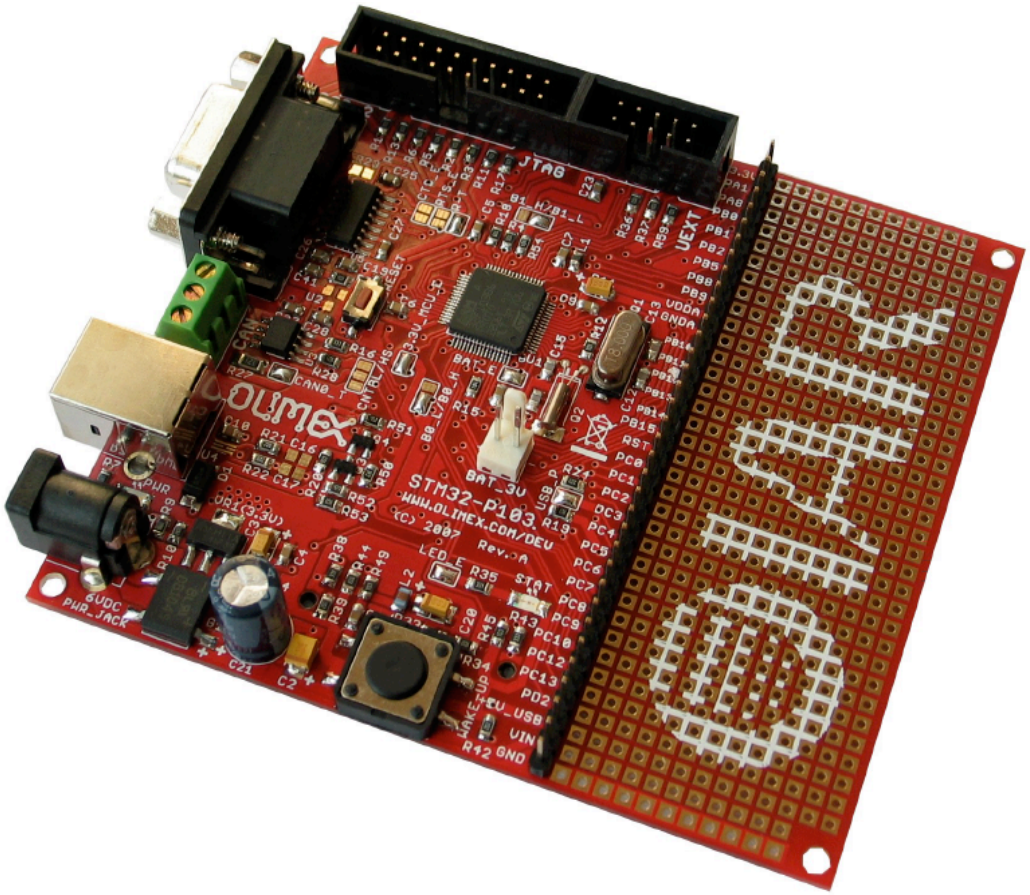


Figura 7 - kit de desenvolvimento STM-P103

3.2.5 Programador/*Debugger*

ST-Link V2 [15] é um programador da ST microeletronics de baixo custo para ser utlizado em conjunto com seus microprocessadores. Este programador da suporte a microcontroladores da familia ARM-32 e ARM-8 da ST. A conexão com o computador é feito através de uma porta USB e como microcontrolador pela interface JTAG. A figura 8 mostra uma foto do programador.

Figura 8 - ST-LINK V2 da STMicroeletronics

3.3 SISTEMA DE COMUNICAÇÃO

3.3.1 STM32F103– Firmware

O kit de desenvolvimento STM32 USB-FS-Device é um firmware completo disponibilizado pela STMicroeletronics para utlizar a inteface USB em full-speed (12Mbps). Este traz exemplos de implementação de todos os tipo de transferências de dados com USB (control, interrupt, bulk and isochronous). Estes exemplos são:

- STM32 USB-FS-Device library: All processes related to default endpoint and standard requests.
- Device firmware upgrade (DFU) demo: Control transfer
- Joystick mouse demo: Interrupt transfer
- Custom HID demo: Interrupt transfer
- Mass storage demo: Bulk transfer
- Virtual COM port: Interrupt and bulk transfer
- USB voice speaker demo (USB speaker): Isochronous transfer
- USB audio streaming demo: Isochronous transfer

Para implementação neste projeto foi utilizado como base o exemplo de Virtual COM port.

3.3.2 Estação-Base- API javax.comm

A API javax.comm oferece uma forma de comunicação serial com a porta Virtual Com em ambiente de programação Java, através dessa API podemos [10]:

- Acessar as portas COM definidas pelo administrador ou usuário.
- Configurar o baud rate, stop bit, e paridade.
- Acessar os buffers de escrita e leitura

Para utilizar a biblioteca Javax.comm no Windows é necessário instalar uma DLL específica e um arquivo properties ambos no diretório Java do computador, ambos estão inclusos no CD do projeto.

3.4 JFREECHART

A biblioteca gráfica utilizada para o desenho do sinal na tela do software da estação base foi a JFreeChart . Está é uma biblioteca “open source” e distribuída sobre sob os termos da GNU Lesser General Public Licence (LGPL), que permite o uso em aplicações proprietárias. [7]

Está API oferece vários tipos de séries de dados e essas são instanciadas através de classes como por exemplo XYSeries. Cada série tem um buffer de dados que são vinculados a um plotador.

A classe XYPlot pode ser utilizada como plotadora, e nesse caso os dados são inseridos aos pares em coodernadas (x,y).

A classe SamplingXYLineRenderer faz o papel de renderização do plotador e sua principal característica é interpolar pontos entre dados inseridos no objeto XYSeries.

4 PROJETO

4.1 DIAGRAMA DE BLOCOS DO SISTEMA

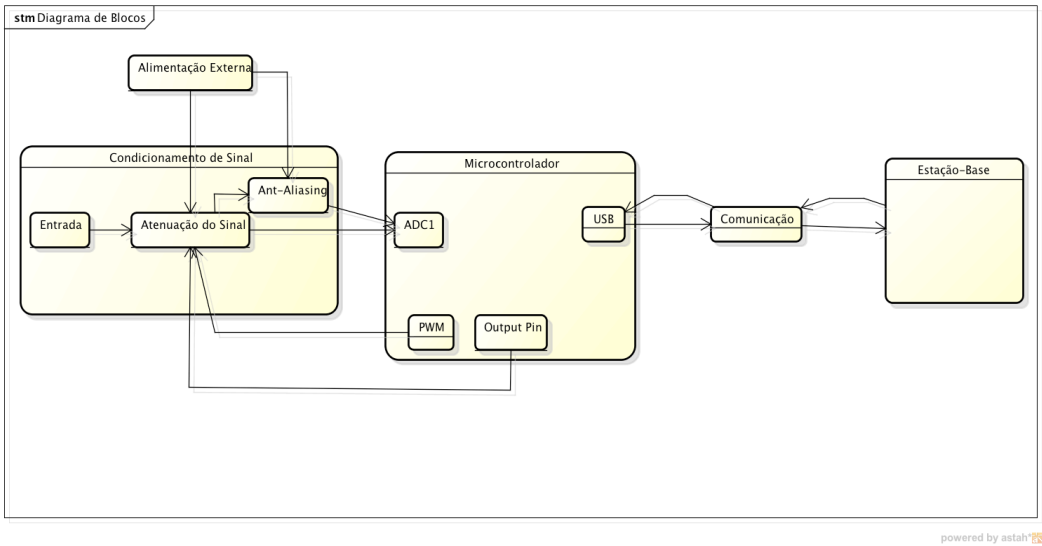


Figura 9 - Visão geral em diagrama de blocos

4.2 CIRCUITO DE CONDICIONAMENTO DE SINAL

O circuito de condicionamento de sinal deveria ter as premissas de fazer atenuações e amplificações do sinal de entrada. E além disso dar suporte a opção da aplicação de um flitro Anti-alising. Para contemplar a premissa de atenuação foi utilizado o conceito de divisores resistivos. A amplificação do sinal de entrada foi feito através da utilização de amplificadores operacionais. E finalmente foi desenvolvido um filtro Butterworth de sexta ordem.

4.2.1 Diagrama Esquemático

O diagrama esquemático foi feito no software Altium Summer Design, pois posteriormente para a fabricação da placa de circuito integrado foi utilizado este mesmo software. A Figura 10 mostra o sistema de aquisição de sinal com todos os componentes utilizados.

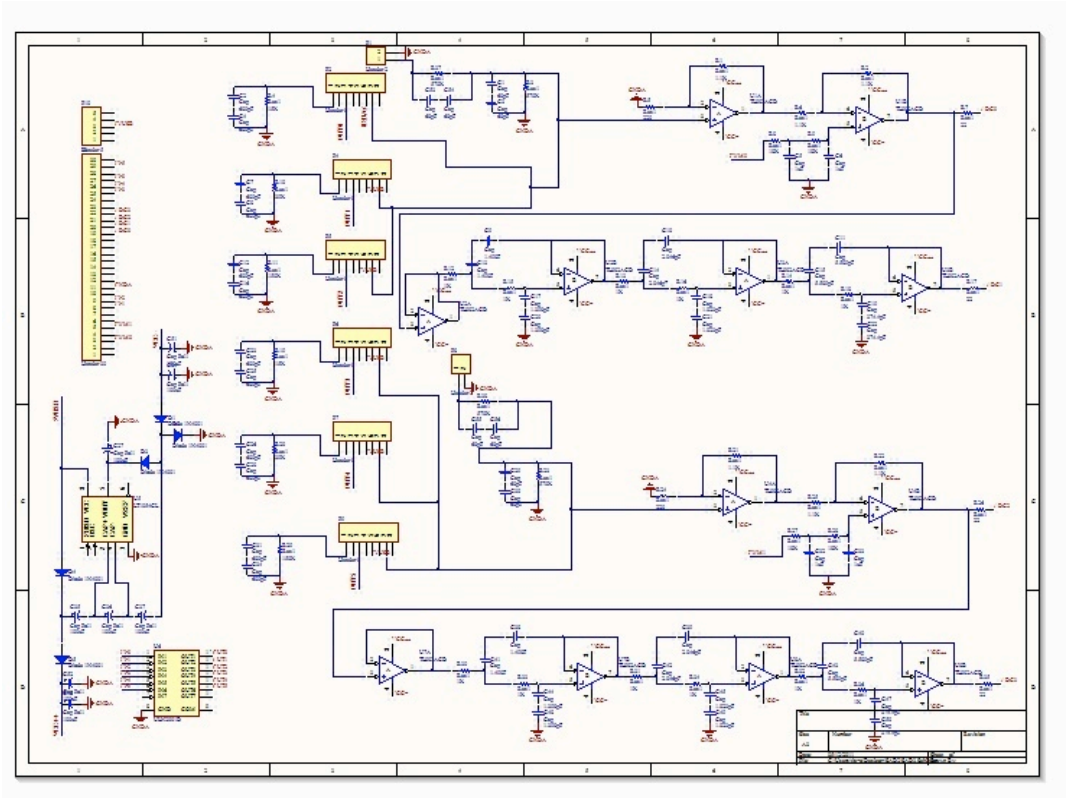


Figura 10 - Sistema de aquisição de sinal

Devido ao grande numero de componentes o esquemático ficou grande e ilegível se visto como um todo. Para sanar tal problema o esquemático será dividido em quatro partes, sendo elas o sistema com o *charge pump* para alimentação dos amplificadores operacionais, o sistema de entrada do primeiro canal, o sistema de entrada do segundo canal e os pinos de conexão com a placa de desenvolvimento.

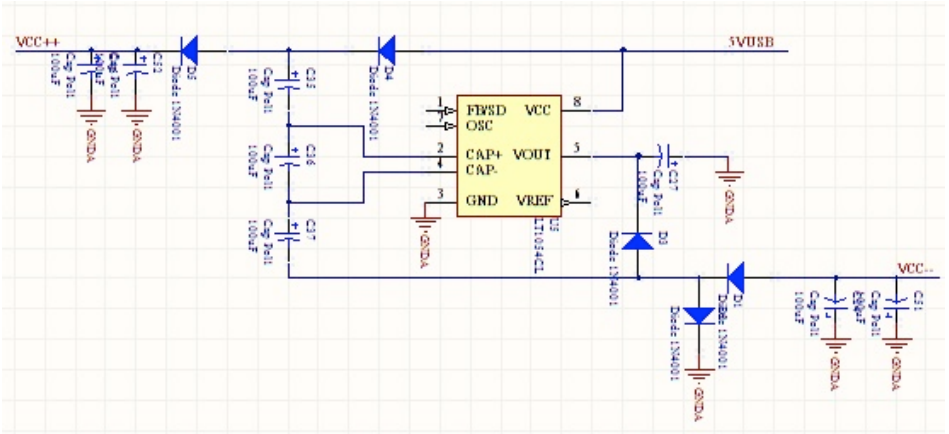


Figura 11 - Sistema de alimentação

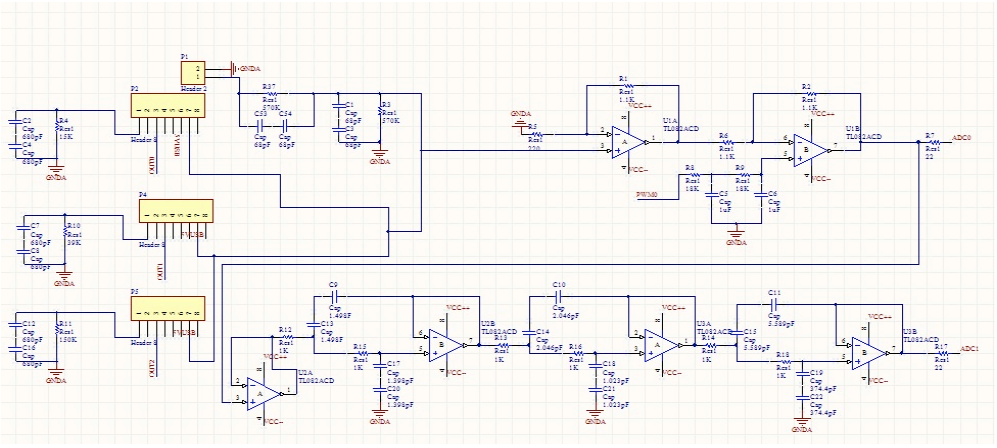


Figura 12 - Circuito de entrada do primeiro canal

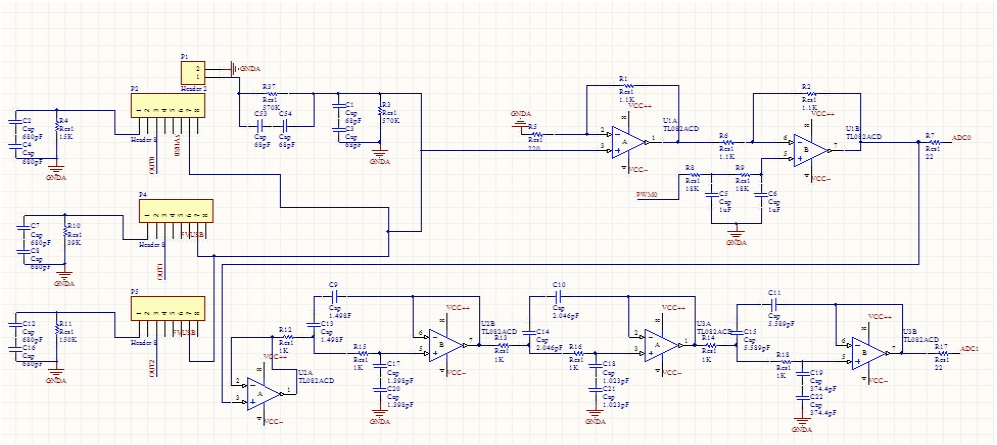


Figura 13 - Circuito de entrada do segundo canal

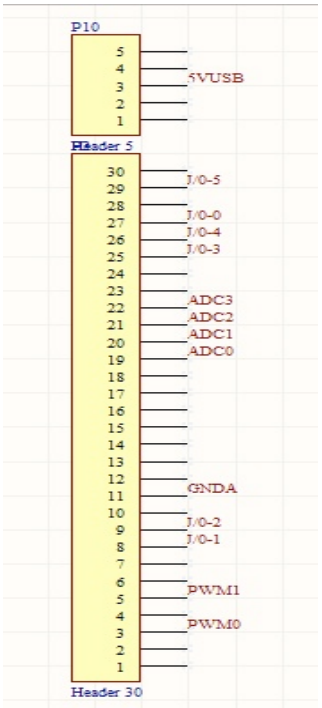


Figura 14 - Pinos de conexão com a placa de desenvolvimento

4.3 PROTOCOLO DE COMUNICAÇÃO

O projeto do protocolo de comunicação foi baseado em características dos protocolos TCP e UDP. Foram aplicadas características do protocolo TCP no envio de comandos de configuração da estação base para o microcontrolador, neste caso necessitamos de uma comunicação confiável, para essa finalidade utilizamos pacotes de reconhecimento (ACK e NACK). Em relação ao protocolo UDP utilizamos suas características para o recebimento de amostras do microcontrolador, nesse caso não precisamos da comunicação confiável, mas sim ocupar banda somente com as amostras enviadas do microcontrolador. Foram utilizadas características do protocolo bit alternante, como o número da janela ser igual a um, ou seja, ao enviar uma mensagem esperamos o ACK (NACK) desta para o envio da próxima mensagem.

A estação base age como mestre nessa comunicação, ela envia comandos de configuração para o microcontrolador. O microcontrolador recebe comandos e envia tanto ACK's como amostras para a estação base, comportando-se, portanto como escravo na comunicação.

No protocolo não utilizamos a abertura de conexão, o microcontrolador sempre está ouvindo a chegada de pacotes de configuração ou requisição de amostras.

Pacotes do protocolo de comunicação

Existem três tipos de pacotes especificados no protocolo: pacote de comando, pacote de dados e pacote de confirmação.

O pacote de comando é enviado pela estação base para o sistema embarcado para modificar as configurações de hardware. A estrutura do pacote é mostrada na Figura 15. Somente os 10 primeiros bits do playload são utilizados. O número do pacote alterna entre 0 e 1, característica equivalente do protocolo bit alternante.

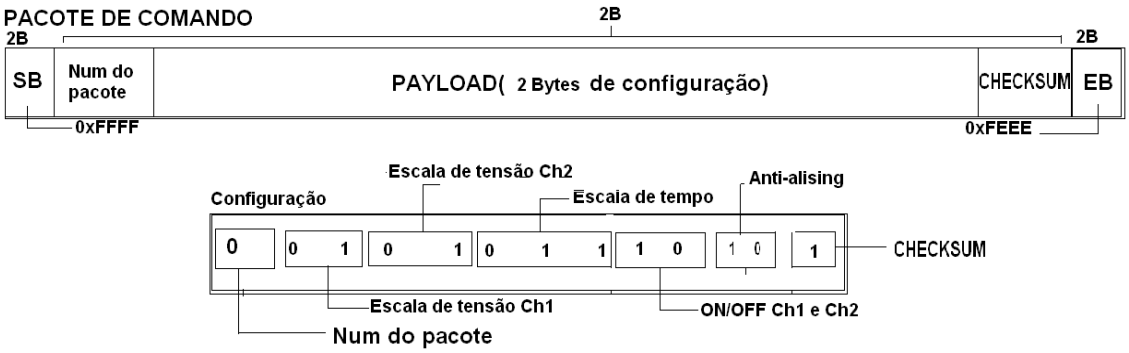


Figura 15 - Pacote de Comando

O pacote de dados consiste em um frame de amostras de um ou dois canais, dependendo da configuração atual. Este pacote é enviado do sistema embarcado para a estação base toda vez que esta faz um requerimento de dados. A Figura 16 mostra a estrutura do pacote de dados. Após o StartByte (SB), é enviado o frame de amostras. Cada amostra contém a informação de qual canal pertence sendo 0x1(canal 1) ou 0x2 (canal 2), em seguida possui 12bits com o valor digital da tensão obtido pelo ADC.

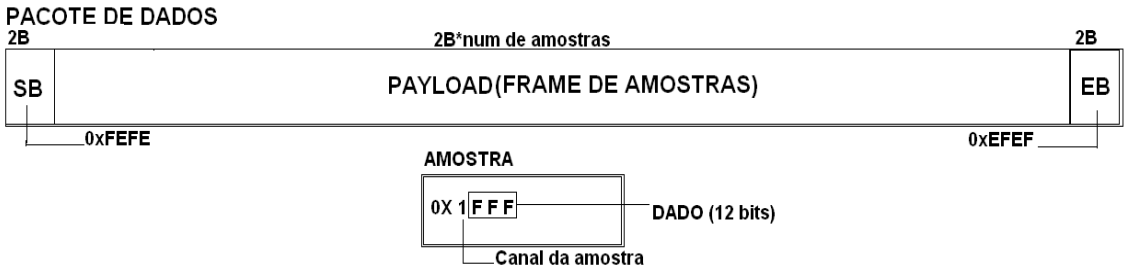


Figura 16 – Pacote de dados

O pacote de confirmação é enviado do sistema embarcado para a estação base confirmando a chegada de comandos. Quando o pacote recebido é reconhecido o pacote de confirmação é do tipo ACK(0), e quando não é reconhecido é do tipo NACK(1). A estrutura do pacote de confirmação é mostrada na Figura 17. O StartByte e o EndByte deste se diferem do pacote de dados para evitar desentendimentos na recepção de

pacotes na estação base. O número do ACK é o mesmo número do pacote de comando que está sendo reconhecido, ou seja, 0x0 ou 0x1.

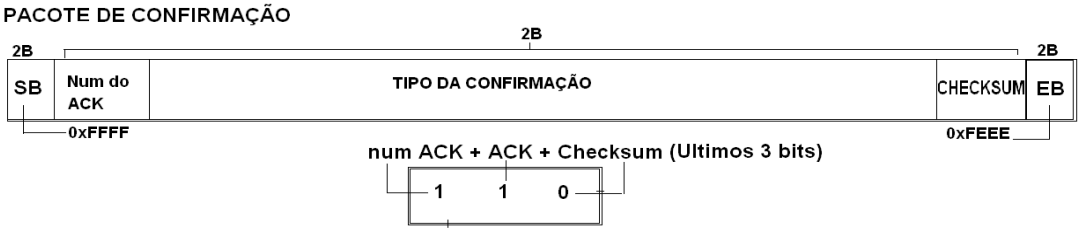


Figura 17 – Pacote de reconhecimento

No pacote de comandos e de confirmação utilizamos o campo Checksum como mecanismo de detecção de erros. No pacote de dados não é necessário retransmissão, portanto este não possui o campo CheckSum.

Houve a necessidade de agrupar os campos de alguns pacotes para gerar um menor *overhead*. Estes agrupamentos podem ser vistos nas imagens dos pacotes.

4.3.1 Diagrama de Máquina de Estados

A estação base após o envio de comandos de configuração ou sempre espera um ACK que confirma a chegada da mensagem no microcontrolador. O funcionamento deste protocolo sem erros em relação à NACK's ou timeout's procede da seguinte forma:

- A conexão é iniciada através do envio de um comando de configuração inicial ao microcontrolador.
- O microcontrolador envia um ACK à estação-base reconhecendo o comando de configuração inicial.
- A estação-base entra no estado esperando evento neste estado tanto comandos de configuração podem ser enviados como amostras podem ser recebidas.
- As amostras quando recebidas são armazenadas e passadas para a camada de aplicação do sistema, que plota as amostras que chegam a partir de outra thread.

- A estação-base só sai desse estado quando envia comandos de configuração ao microcontrolador. E quando sai desse estado não recebe amostras de dados dos canais, somente ao receber o ACK do comando enviado, impedindo que os dados plotados sejam diferentes da configuração que o usuário selecionou.

  Outros pontos importantes
- Na estação-base, na espera de um ACK de comando enviado, pode ocorrer TimeOut deste. Caso a contagem de timeout's ultrapasse um determinado valor máximo a conexão será encerrada, caso contrário ocorre a retransmissão do comando.
- O mesmo vale para NACK's enviados do microcontrolador, essa medida foi tomada caso a conexão fique muito ruidosa e, portanto, a estação-base receberia muitos NACK's de comandos enviados e o usuário não teria nenhum conhecimento do que estivesse realmente acontecendo no sistema.

4.3.1.1 Estação Base

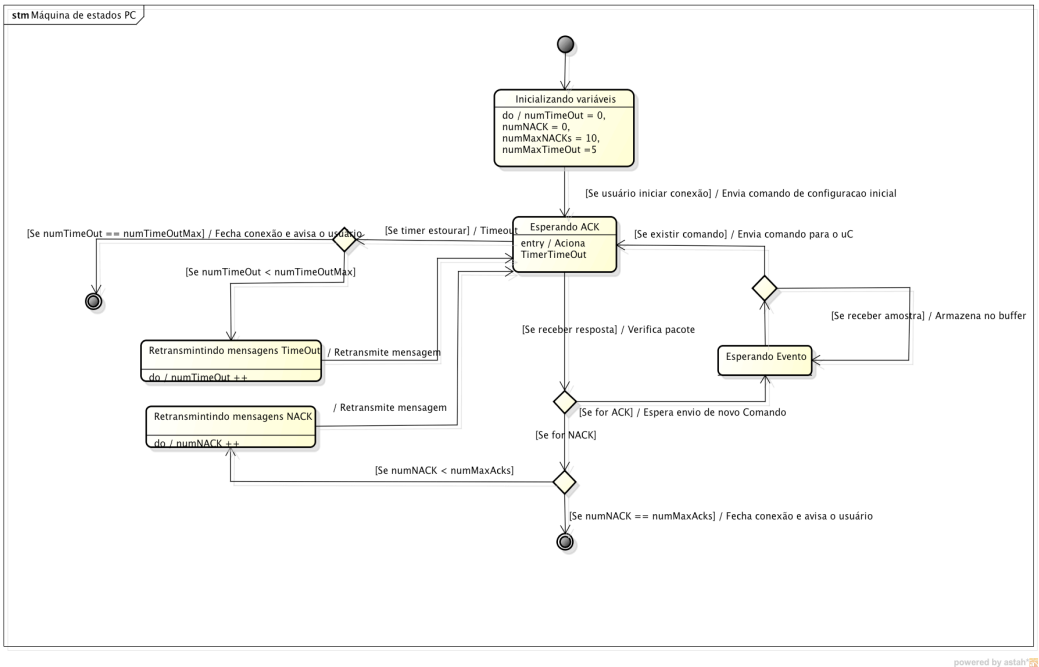

stm Máquina de estados PC
Inicializando variáveis
do / numTimeOut = 0,
numNACK = 0,
numMaxNACKs = 10,
numMaxTimeOut =5
[Se usuário iniciar conexão] / Envia comando de configuracao inicial
Esperando ACK
entry / Aciona
TimerTimeOut
[Se numTimeOut == numTimeOutMax] / Fecha conexão e avisa o usuário
[Se timer estourar] / Timeout
[Se existir comando] / Envia comando para o uC
[Se numTimeOut < numTimeOutMax]
[Se receber amostra] / Armazena no buffer
[Se receber resposta] / Verifica pacote
Retransmintindo mensagens TimeOut
/ Retransmite mensagem
do / numTimeOut ++
Esperando Evento
Retransmintindo mensagens NACK
/ Retransmite mensagem
do / numNACK ++
[Se for ACK] / Espera envio de novo Comando
[Se for NACK]
[Se numNACK < numMaxAcks]
[Se numNACK == numMaxAcks] / Fecha conexão e avisa o usuário

Figura 18 – Protocolo de comunicação estação base

4.3.1.2 Sistema Embarcado

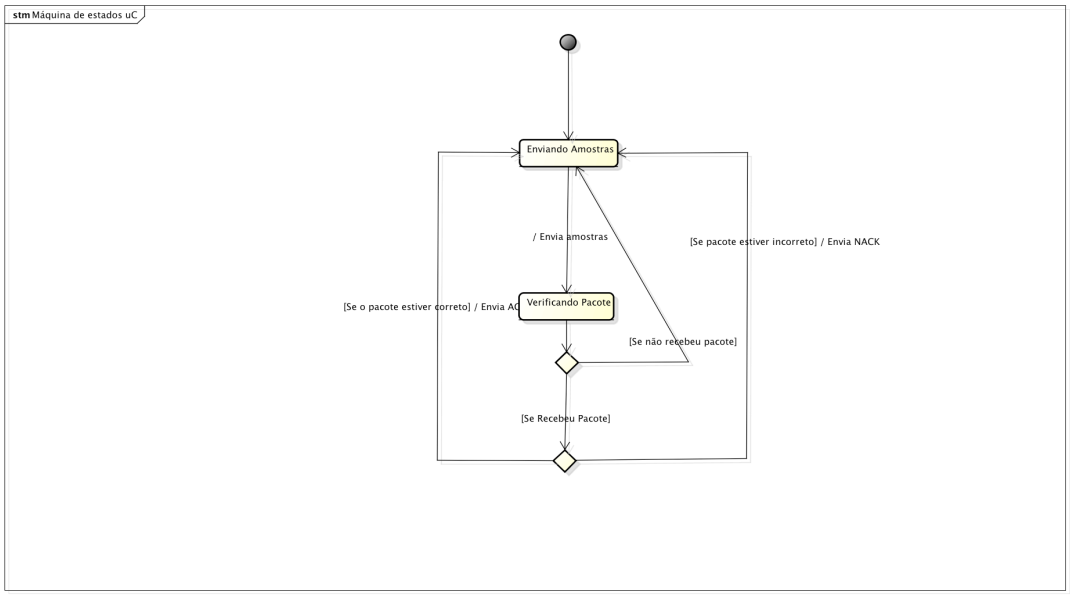

stm Máquina de estados uC
Enviando Amostras
/ Envia amostras
[Se pacote estiver incorreto] / Envia NACK
Verificando Pacote
[Se o pacote estiver correto] / Envia AC
[Se não recebeu pacote]
[Se Recebeu Pacote]
powered by astah*

Figura 19 - Protocolo de comunicação sistema embarcado

4.4 SISTEMA EMBARCADO

Nesta sessão será apresentado o projeto do software embarcado do osciloscópio digital USB. O projeto consiste dois diagramas de estados e suas respectivas descrições.

4.4.1 Diagrama de Máquina de Estados

Os diagramas a seguir mostra a máquina de estados do software do sistema embarcado.

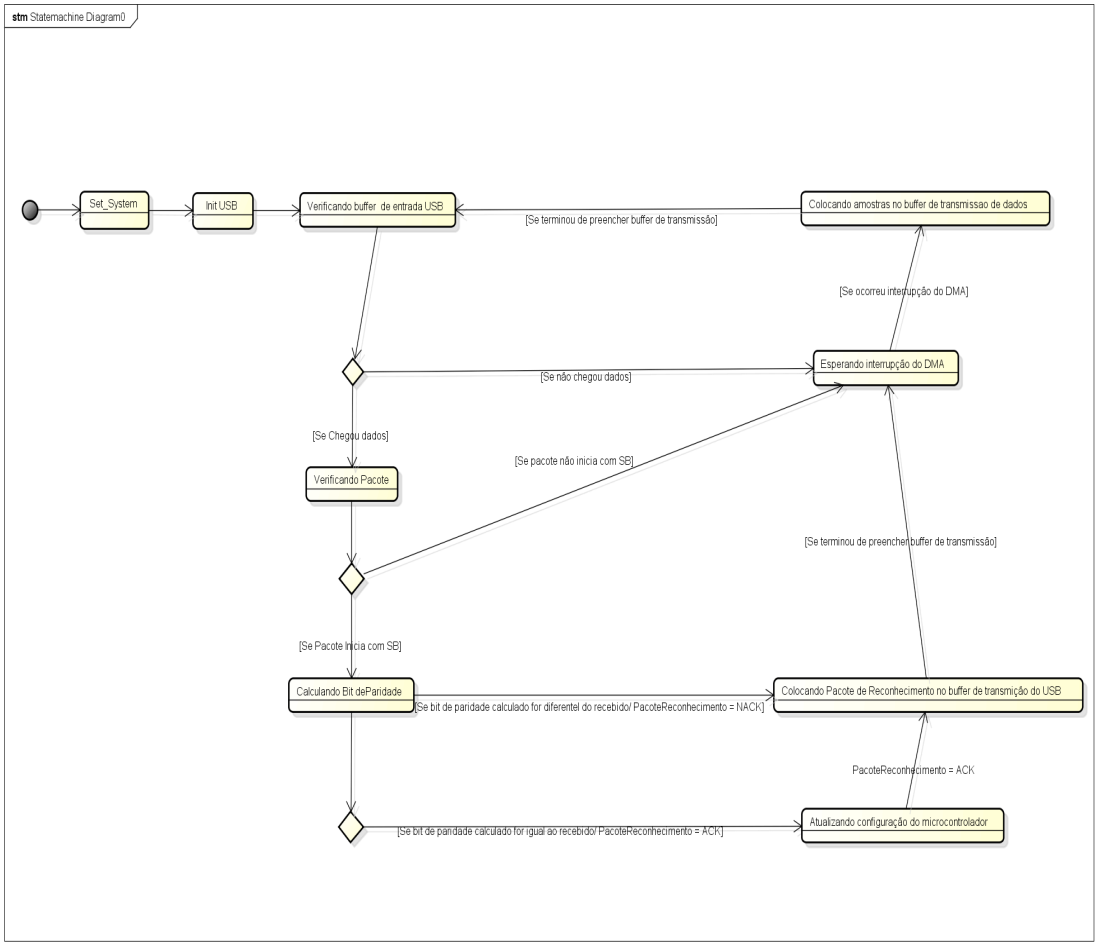

stm Statemachine Diagram0
Set_System
Init USB
Verificando buffer de entrada USB
[Se terminou de preencher buffer de transmissão]
Colocando amostras no buffer de transmissao de dados
[Se ocorreu interrupção do DMA]
[Se não chegou dados]
Esperando interrupção do DMA
[Se Chegou dados]
[Se pacote não inicia com SB]
Verificando Pacote
[Se terminou de preencher buffer de transmissão]
[Se Pacote Inicia com SB]
Calculando Bit deParidade
[Se bit de paridade calculado for diferentel do recebido/ PacoteReconhecimento = NACK]
Colocando Pacote de Reconhecimento no buffer de transmição do USB
PacoteReconhecimento = ACK
[Se bit de paridade calculado for igual ao recebido/ PacoteReconhecimento = ACK]
Atualizando configuração do microcontrolador

Figura 20 – Diagrama de Máquina de Estados do Sistema Embarcado

Nos estados Set_System e Init_USB São feitas as configurações iniciais do microcontrolador e de seus periféricos após isso a máquina fica em loop infinito nos seguintes passos.

- Verifica se existe pacote enviado pela estação base no buffer de entrada (SB)
    - Se existir verifica pacote (Checksum)
        - Se estiver correto envia pacote de confirmação ACK
        - Se estiver incorreto envia pacote de confirmação NACK.
- Espera interrupção do DMA
- Quando ocorrer interrupção envia buffer de dados para a estação base

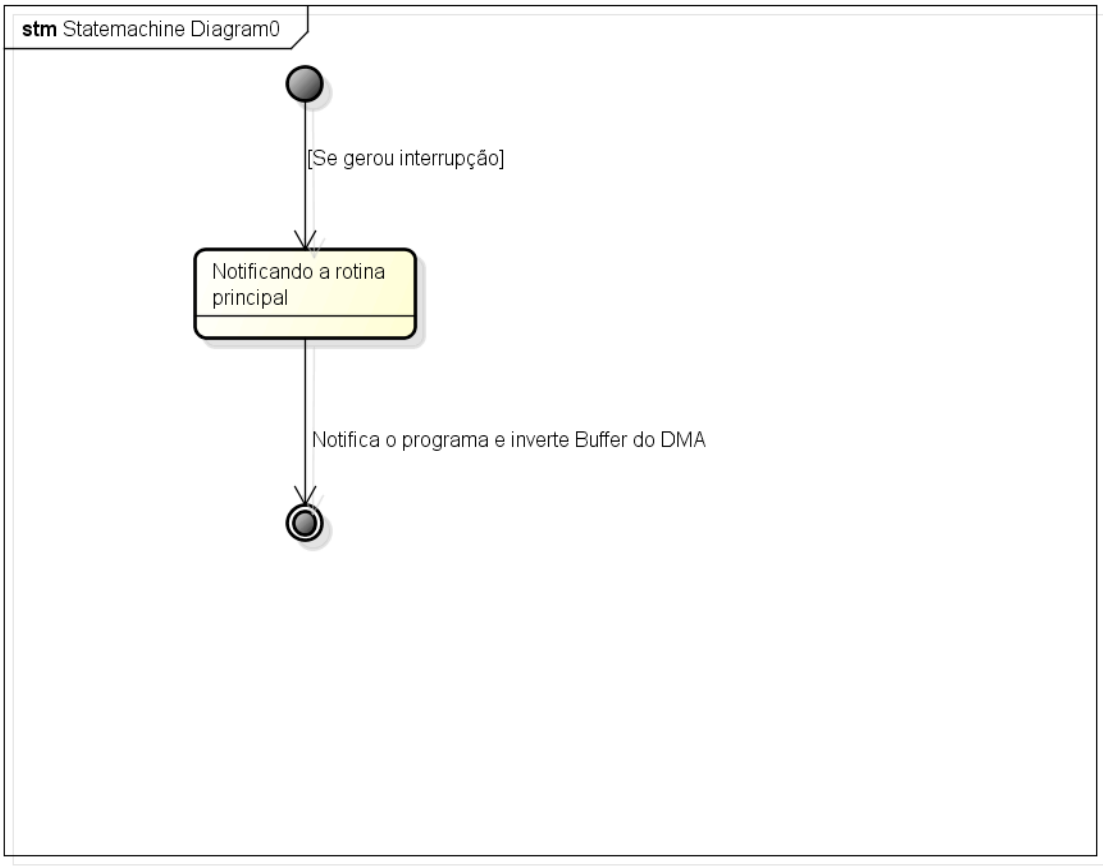

stm Statemachine Diagram0
[Se gerou interrupção]
Notificando a rotina principal
Notifica o programa e inverte Buffer do DMA

Figura 21 - Diagrama 2 de Máquina de Estados do Sistema Embarcado

Toda vez que ocorre uma interrupção do DMA sinalizando que foi convertido um buffer de dados pelo ADC o tratador faz os seguintes passos:

- Ativa uma flag para sinalizar para a rotina principal que existem dados no buffer para serem enviados para a estação base.
- Troca o buffer que o DMA irá armazenar as amostras (dual buffer).

4.5 ESTAÇÃO BASE

Nesta sessão é apresentado todos projetos referentes ao software da estação base, sendo eles, diagrama de casos de uso, diagrama de classes e diagrama de sequencia.

4.5.1 Diagrama de Casos de Uso

O diagrama de casos de uso é um diagrama da UML cujo objetivo é representar um requisito do sistema que será automatizado. A Figura 22 mostra o diagrama de casos de uso para o software da estação base, logo abaixo da figura segue sua descrição.

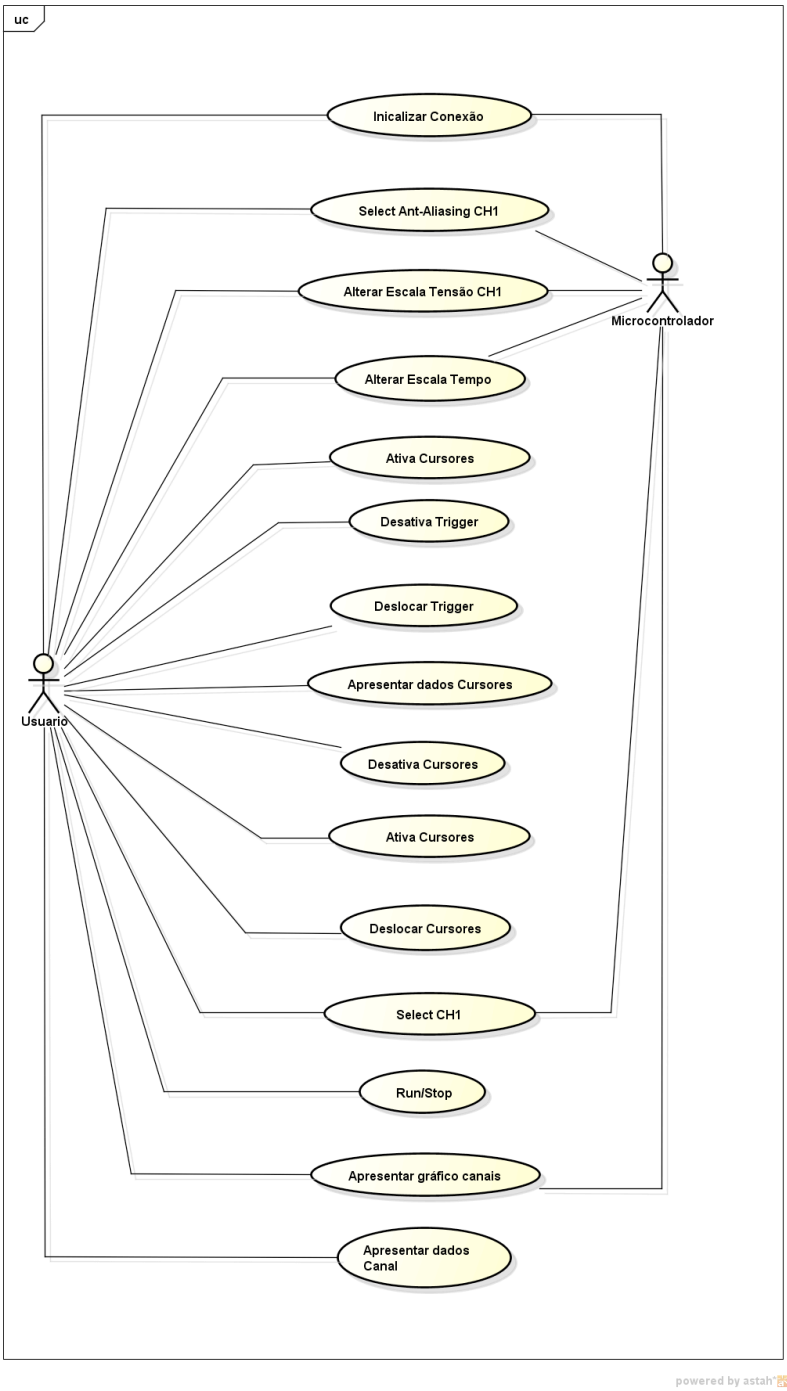


Figura 22 - Diagrama de casos de uso

- UC01: Inicializar conexão

  O usuário inicia a conexão USB com o sistema controlado pelo microcontrolador.

- UC02: Select Anti-Aliasing CH1

  O usuário pode ativar ou desativar a opção de Anti-Aliasing do sinal obtido no canal 1.

- UC03: Alterar Escala de Tensão CH1

  O usuário pode aumentar ou diminuir a escala de tensão através de escalas definidas no software para o sinal obtido no canal 1 .

- UC04: Alterar Escala de Tempo

  O usuário pode aumentar ou diminuir a escala de tempo do sinal obtido.

- UC05: Ativar Cursores

  O usuário pode ativar a visualização dos cursores vertical do sinal obtido.

- UC06: Desativar Trigger

  O usuário pode desativar o trigger em relação ao sinal obtido.

- UC07: Deslocar Trigger

  O usuário pode deslocar a possição do trigger.

- UC08: Apresentar dados Cursores

  É apresentado os dados respctivos aos cursores verticais caso estejam ativos.

- UC09: Desativar Cursores

  O usuário pode desativar a visualização dos cursores vertical do sinal obtido.

- UC10: Deslocar Cursores

  O usuário pode deslocar a possição dos cursores verticais.

- UC11: Select CH1

  O usuário pode ativar visualização do canal 1.

- UC12: Run/Stop

  O usuário pode parar a captura do sinal ou continuar a capturar nos dados.
- UC13: Apresentar Dados Canail

  É apresentado ao usuário os dados obtidos nos canais ativos.
- UC14: Apresentar Gráfico Canais

  É apresentado ao usuário um gráfico contendo o sinal obtido.

4.5.2 Diagrama de Classes

A partir do diagrama de casos de uso, foi possivel o desenvolvimento de mais dois diagramas, classes e sequencia. O diagrama de classes que é apresentado na Figura 23.

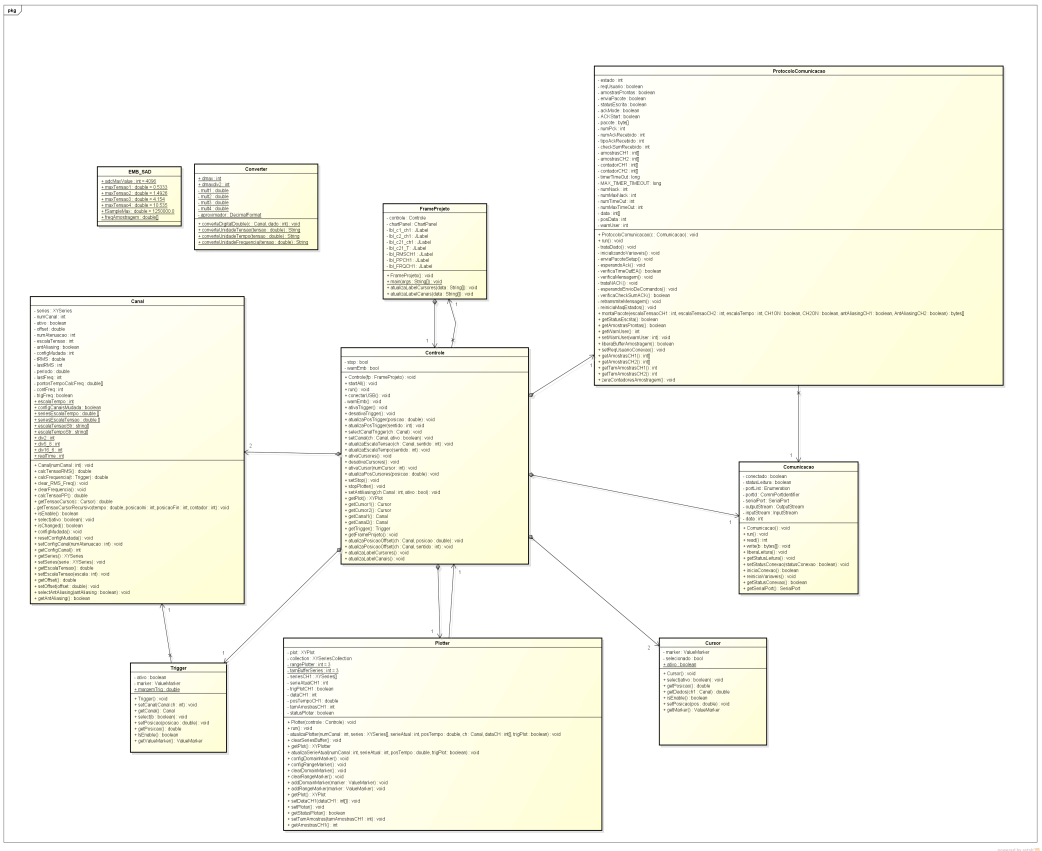


Figura 23 - Diagrama de Classes

- **Public class FrameProjeto**

extends JFrame

Essa classe possuirá todos os elementos da interface do osciloscópio, como botões,
labels, painéis, etc. Em seu construtor um objeto Controle é instanciado.

- **Public class Controle**

implements Runnable

Essa classe fará o controle do software da estação base em relação à comunicação do sistema embarcado e a estação base analisando pacotes que chegam do sistema embarcado, e a escrita de pacotes para o sistema embarcado. A classe também faz a ligação entre a interface com as classes: Canal, Cursor, Plotter e Comunicação, além de possuir as variáveis estáticas de configuração, que afetam os gráficos plotados.

- **Public class Comunicação**

implements Runnable

Classe que representa a comunicação USB da estação base.

- **Public class Plotter**

implements Runnable

O Plotter será a classe que representa o gráfico do osciloscópio. As plotagens de tensão em relação ao tempo serão geradas continuamente. Ele utilizará as flags de controle da classe Controle com o objetivo de controlar essas plotagens.

- **Public class Cursor**

Essa classe foi criada para representar os cursores do osciloscópio. Os cursores serão linhas verticais que percorrem o plotador e, portanto, sua posição varia no domínio do tempo.

- **Public class Canal**

Essa classe foi criada para representar os canais do osciloscópio.

4.5.3 Diagramas de Sequência

O diagrama de sequência finaliza a etapa de projeto de software, o mesmo está descrito em seguida.

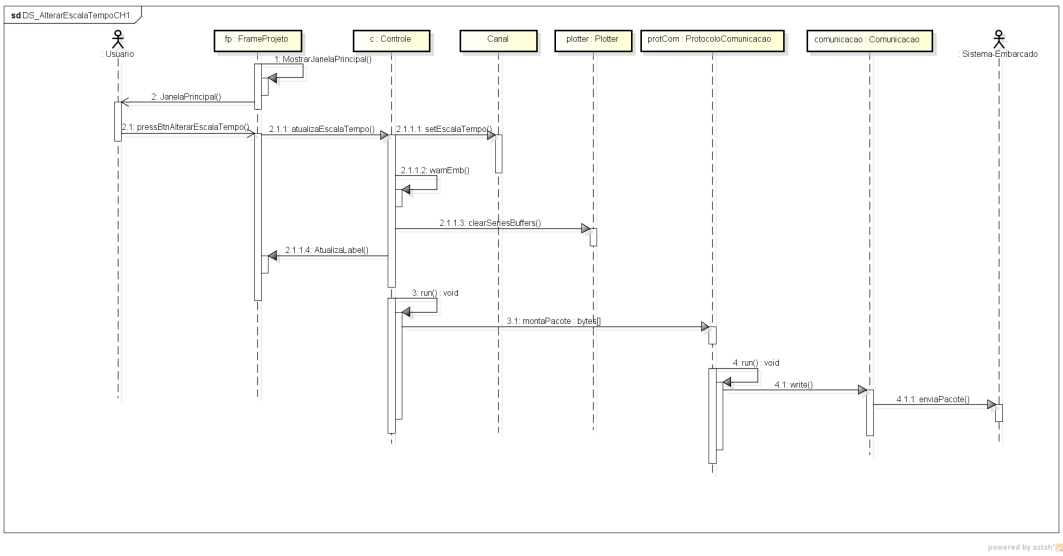


Figura 24 – Diagrama de sequência para alterar escala de tempo do canal 1

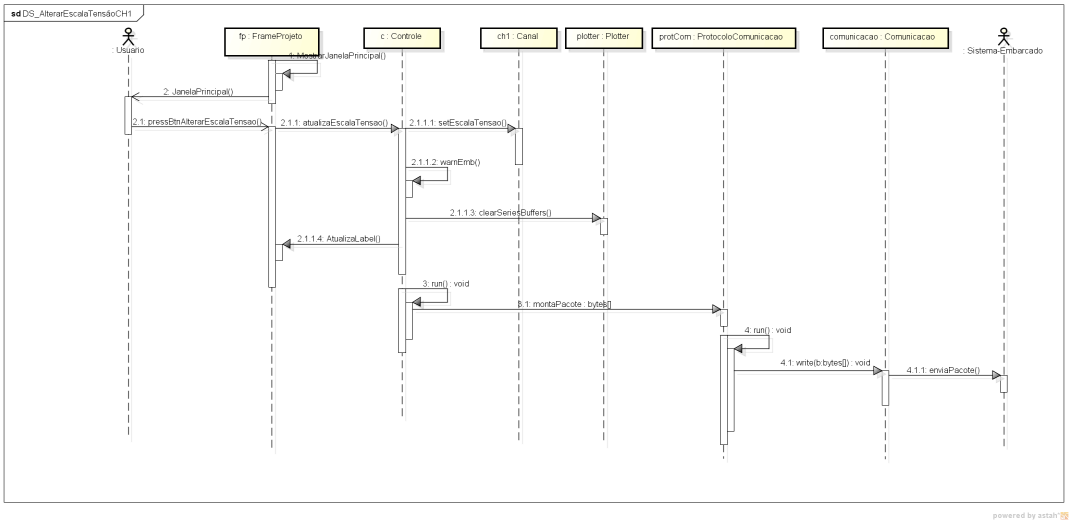


Figura 25 – Diagrama de sequência para alterar escala de tesão do canal 1

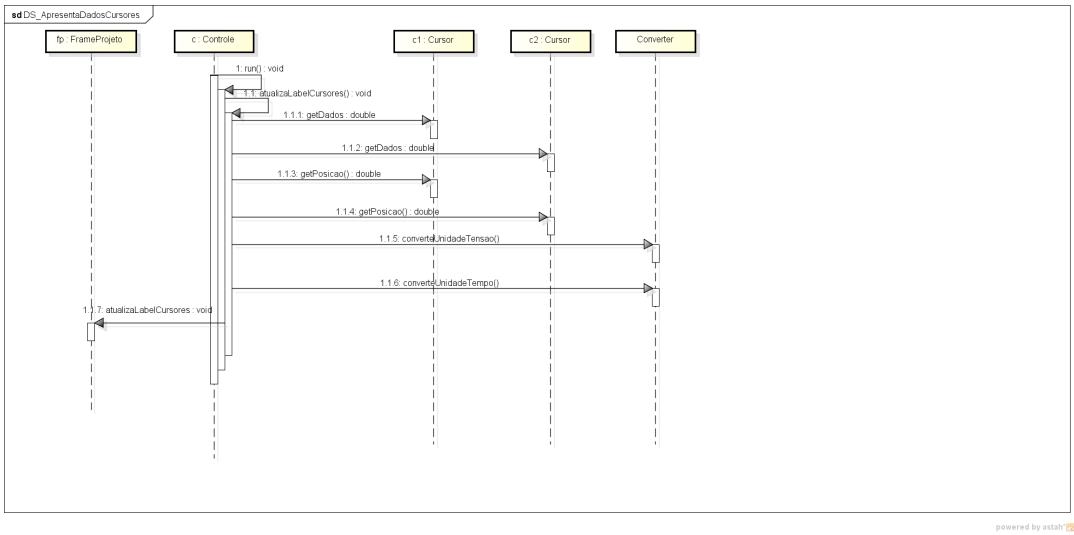


Figura 26 – Diagrama de sequência para mostrar dados dos cursores

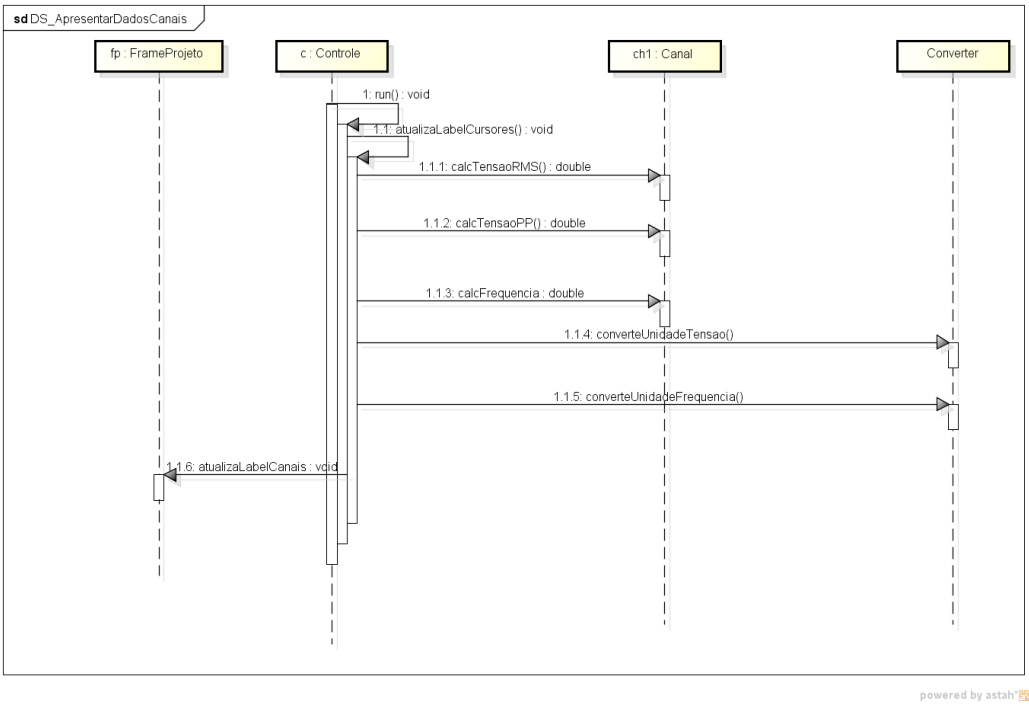


Figura 27 – Diagrama de sequência para apresentar dados dos canais

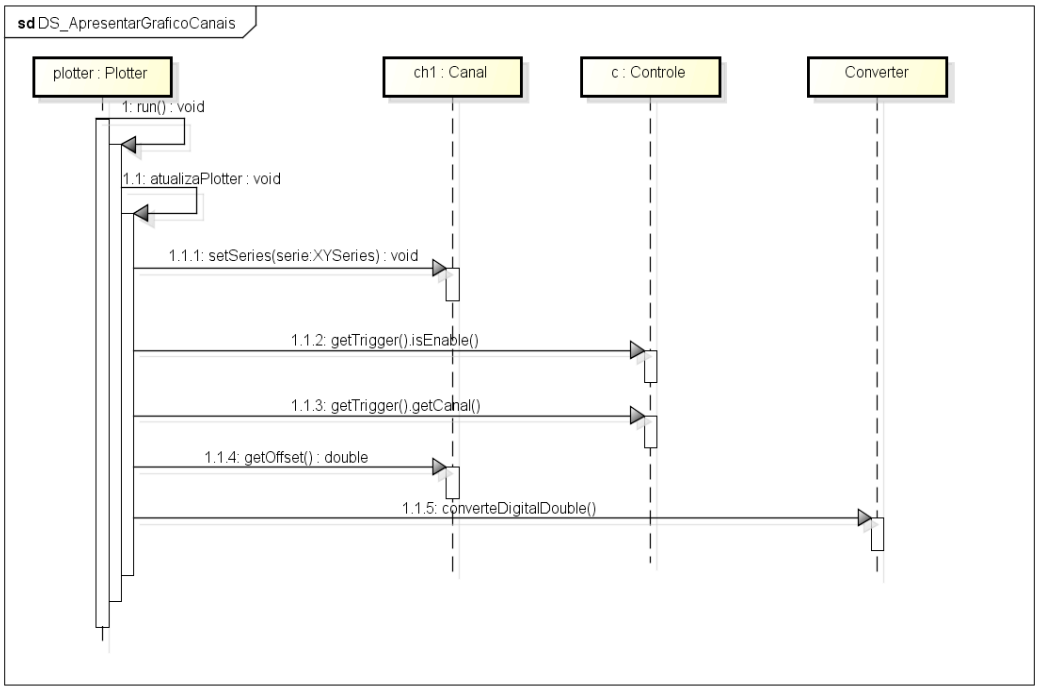


Figura 28 – Diagrama de sequência para apresentar o gráfico dos canais

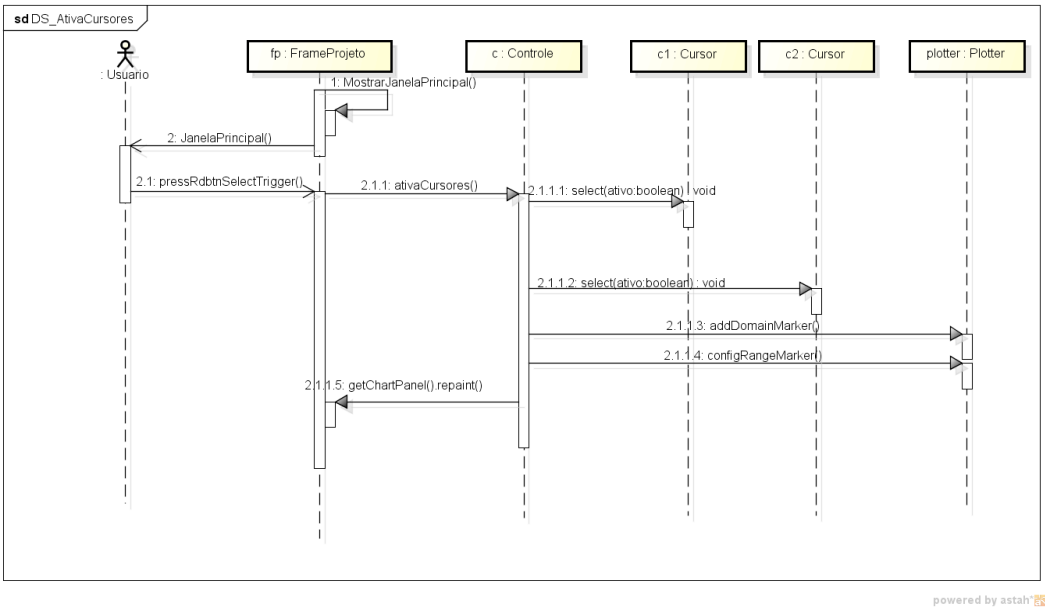


Figura 29 – Diagrama de sequência para ativar cursores

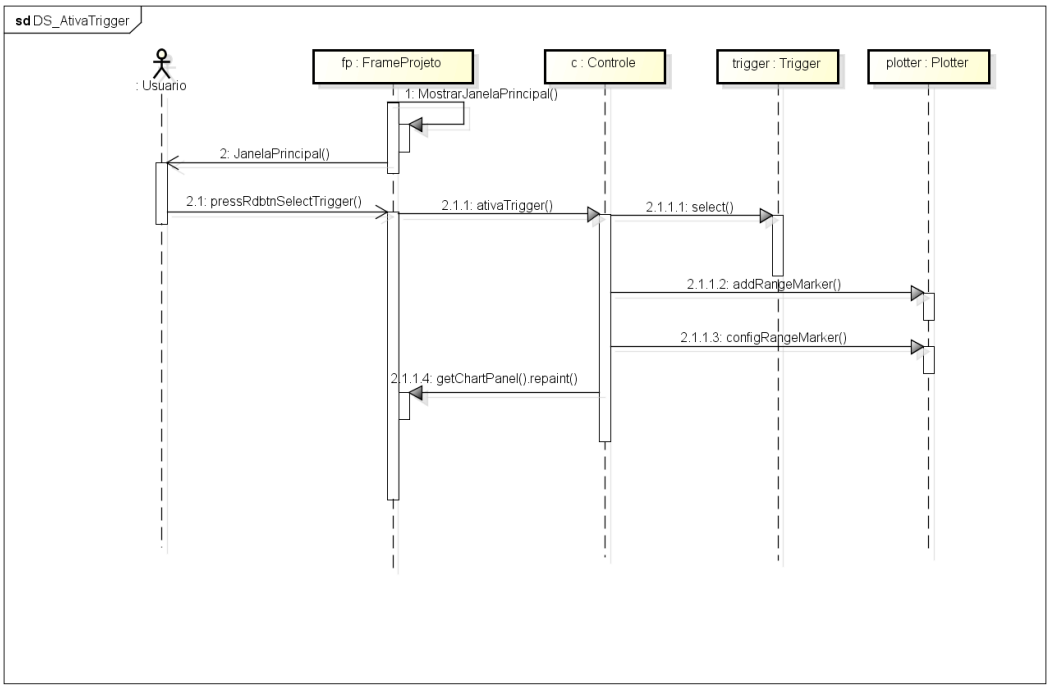


Figura 30 – Diagrama de sequência para ativar o trigger

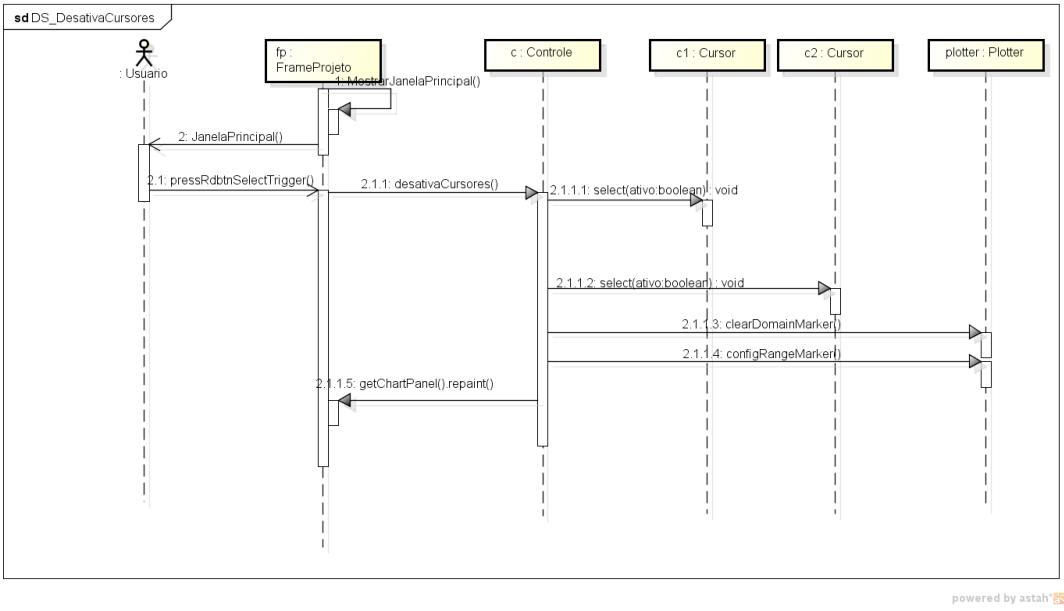


Figura 31 – Diagrama de sequência para desativar os cursores

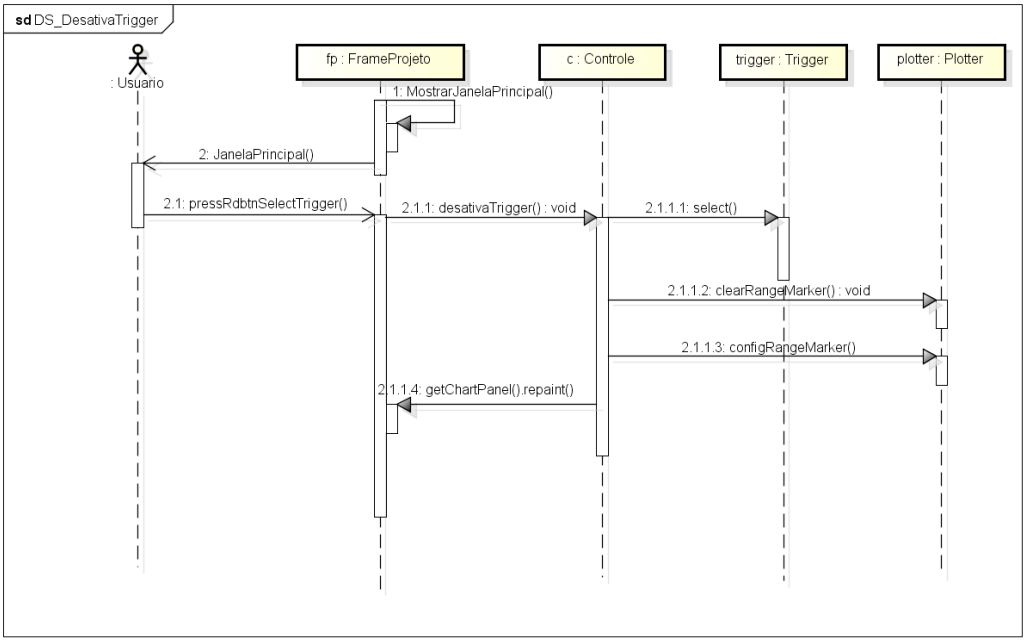


Figura 32 – Diagrama de sequência para desativar o trigger

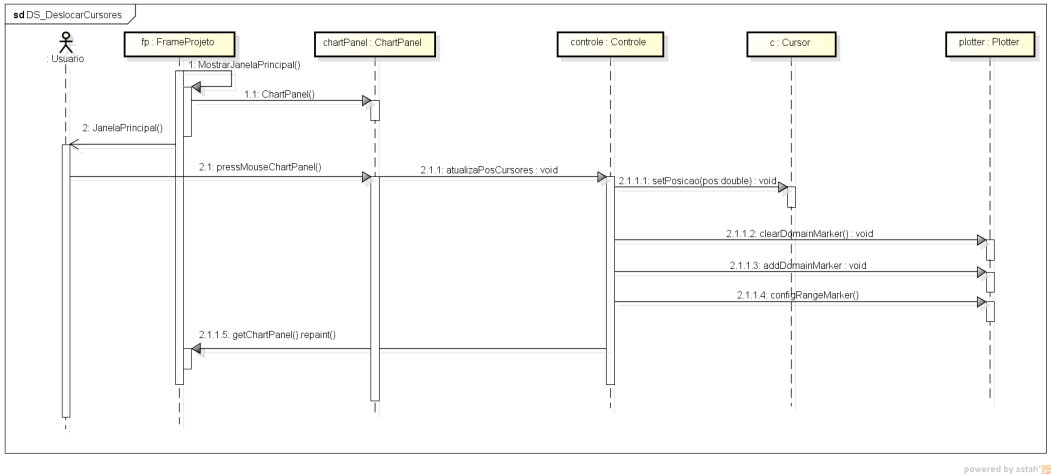


Figura 33 – Diagrama de sequência para deslocar os cursores

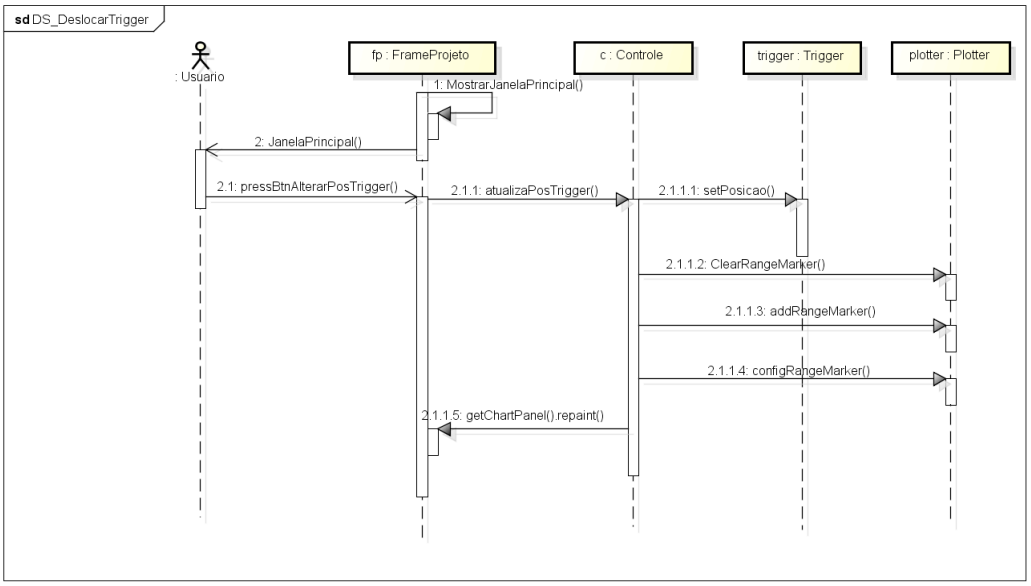


Figura 34 – Diagrama de sequência para deslocar o trigger

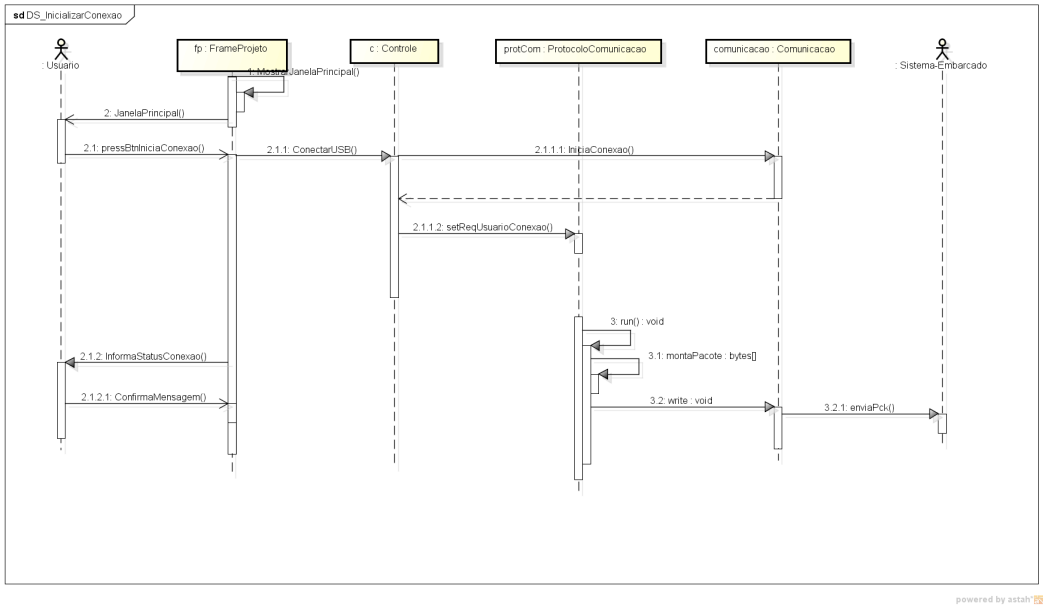


Figura 35 – Diagrama de sequência para iniciar conexão

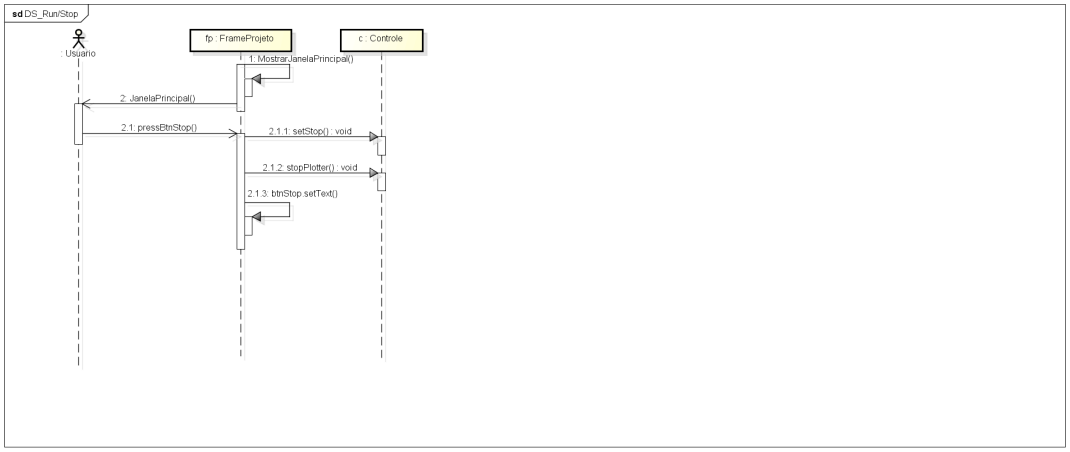


Figura 36 – Diagrama de sequência para eventos Run e Stop

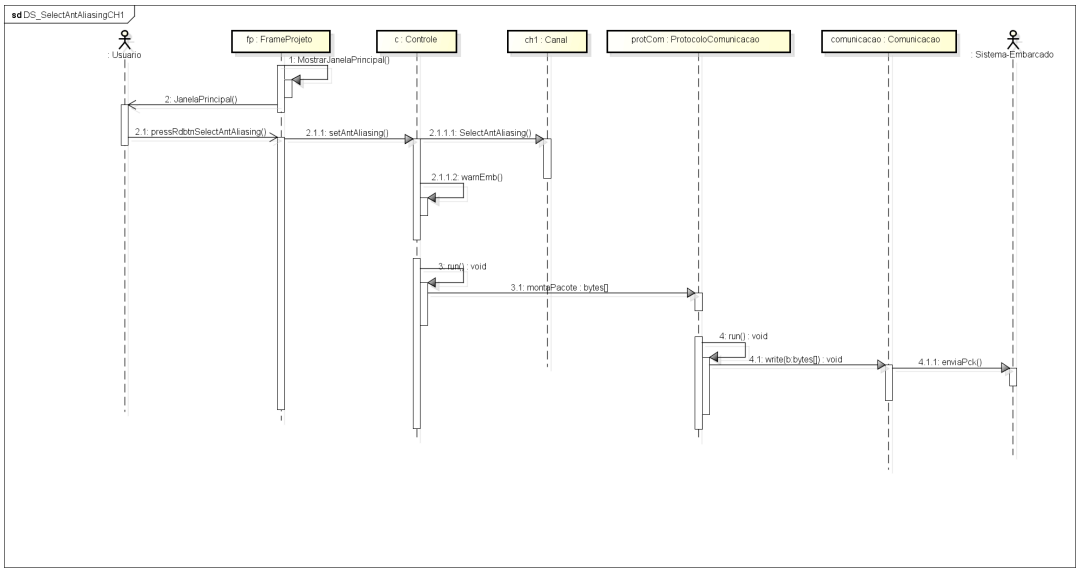


Figura 37 – Diagrama de sequência para selecionar o Anti-Aliasing no canal1

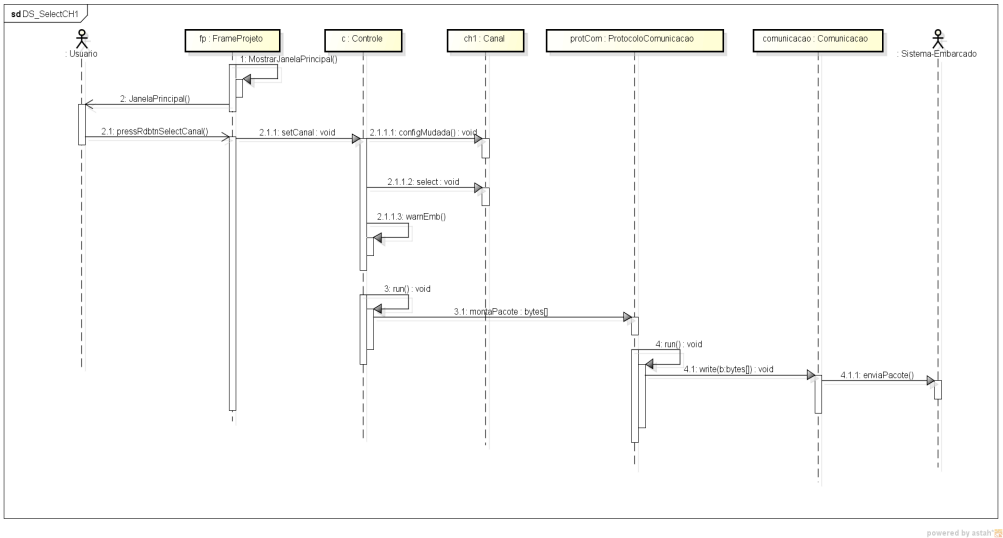


Figura 38 – Diagrama de sequência para desativar o Anti-Aliasing do canal1

4.6 PLACA DE CIRCUITO IMPRESSO

Para o desenvolvimento das placas de circuito impresso (PCI), foi utilizado o software Altium [9]. Segue os layout das PCIs:

4.6.1 Placa do Circuito de Condicionamento de Sinal

Foi necessário a confecção de uma placa dupla face devido ao grande numero de ligações e componentes. As trilhas foram feitas com a espessura de 14 milímetros e, além disso, foram tomados cuidados como não deixar trilhas muito perto uma das outras nem trilhas muito perto dos furos dos componentes e também fazer trilhas com o menor comprimento possível. Todos esses cuidados foram tomados com o objetivo de se diminuir o risco de curtos ou trilhas “quebradas” no estagio de fabricação. Abaixo segue as figuras da face superior e inferior consecutivamente.

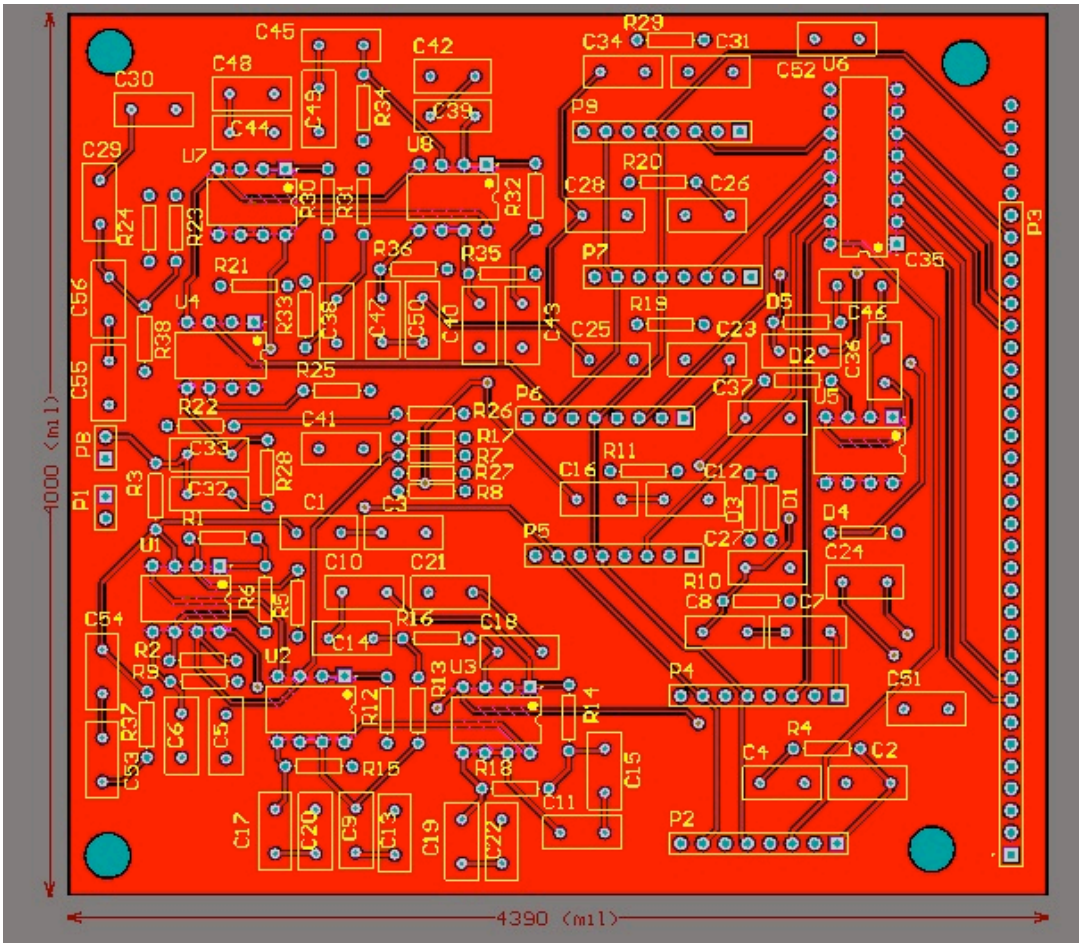


Figura 39 - Face superior da PCI do SAD

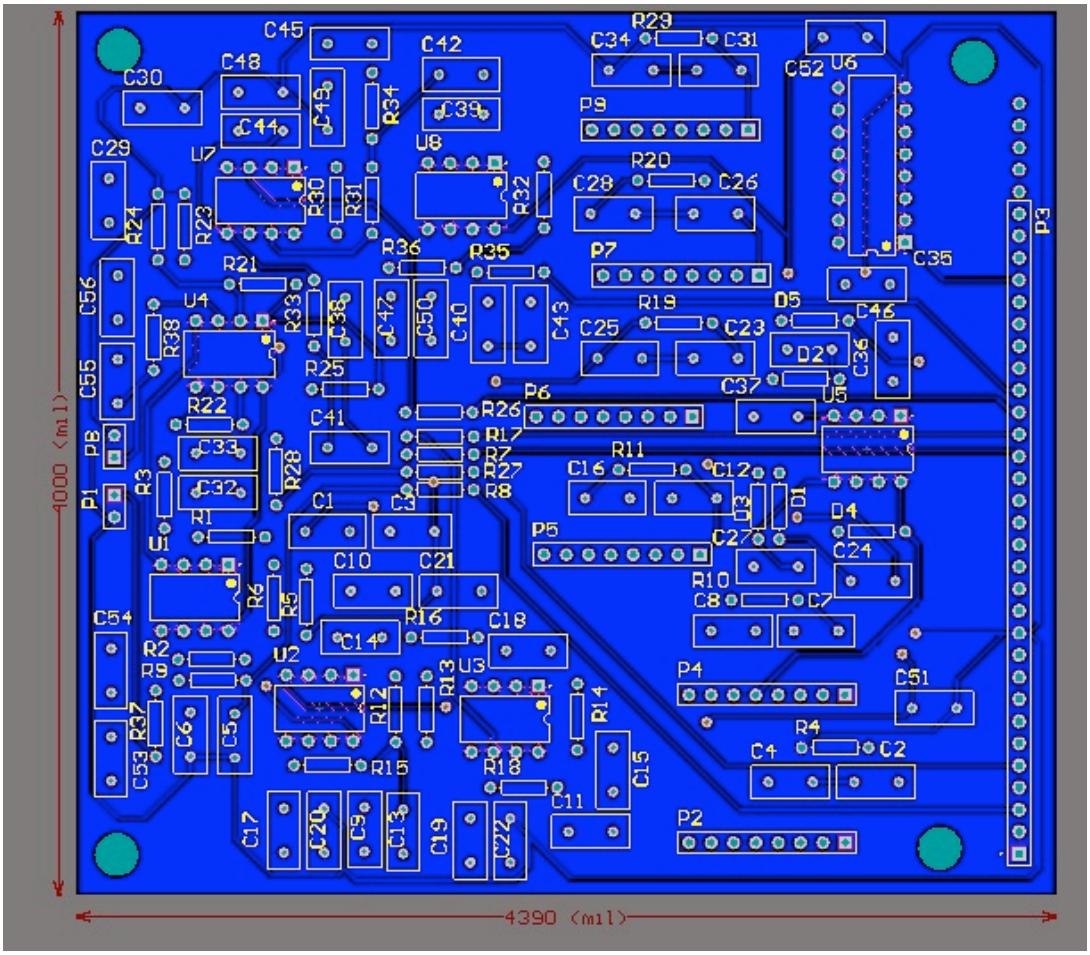


Figura 40 - Face inferior da PCI do SAD

4.6.2 Placa do Circuito de Condicionamento de Sinal com Microcontrolador Integrado

Nesta placa foram utilizados componentes DIP e componentes SMD. Para esta placa foi necessário utilizar trilhas menores com a espessura de 6 milímetros devido ao encapsulamento do microcontrolador. Esta placa também é uma placa dupla face. Foram tomados os mesmo cuidados da placa de condicionamento de sinal para diminuir os riscos de erros no estagio da fabricação. Abaixo segue figuras da face superior e face inferior consecutivamente.

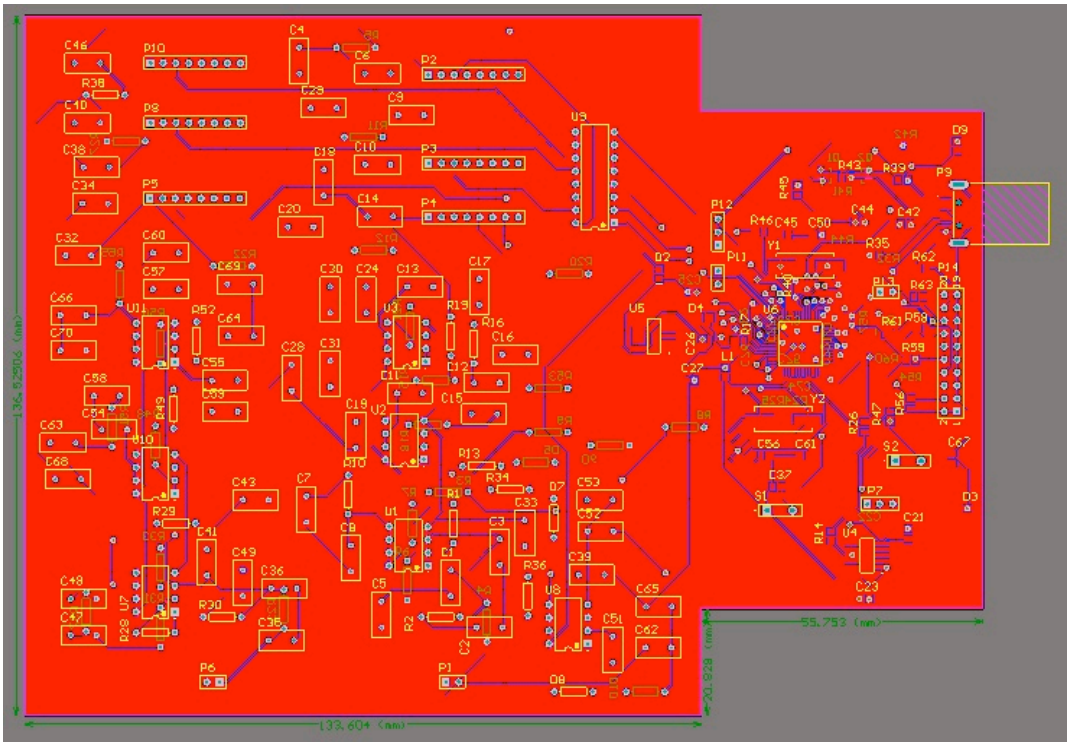


Figura 41 - Face superior da PCI do microcontrolador

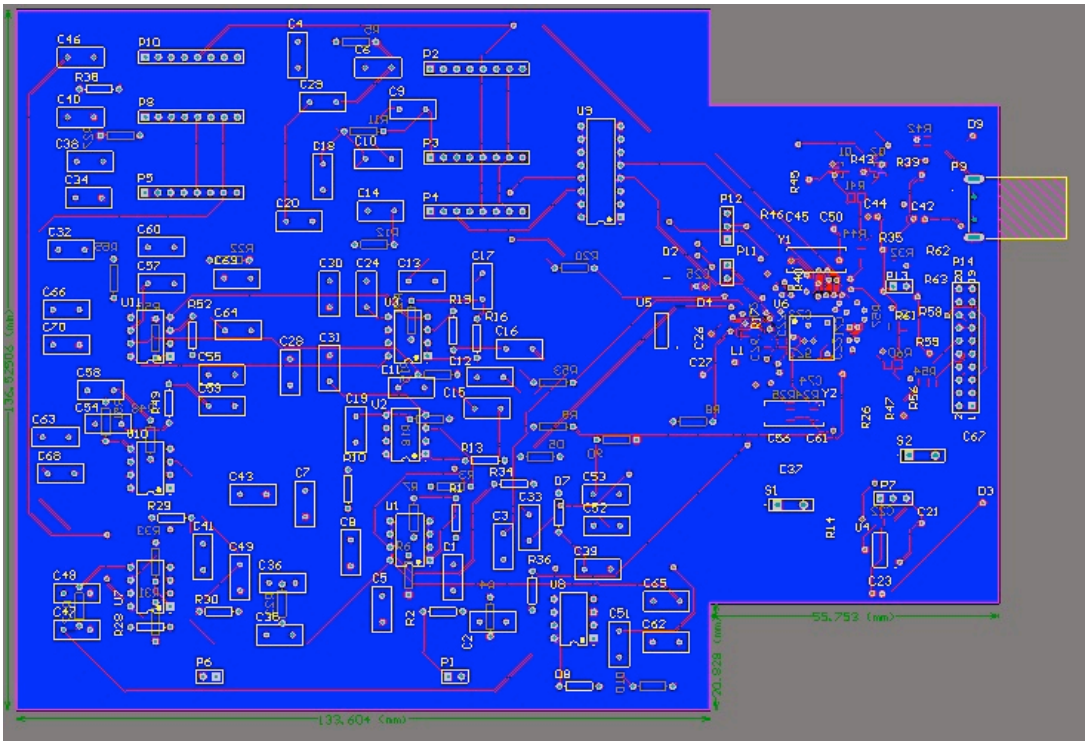


Figura 42 - Face inferior da PCI do microcontrolador

5 EXECUÇÃO

5.1 COMPONENTES UTILIZADOS

Os componentes utilizados para o desenvolvimento da PCI de sistema de aquisição de sinais (SAD) foram os seguintes:

- Resistores
  - 10 de 1000Ω;
  - 5 de 22Ω;
  - 5 de 15KΩ;
  - 5 de 39KΩ;
  - 5 de 150KΩ;
  - 5 de 220Ω;
  - 5 de 18KΩ;
  - 5 de 1.1KΩ;
- 1 Charge Pump
- Capacitores
  - 4 de 1,4nF
  - 4 de 1,3nF
  - 4 de 2nF
  - 4 de 1nF
  - 4 de 4,7nF
  - 4 de 374,4pF
  - 4 de 1,023 pF
  - 4 de 2,046 pF
  - 4 de 1,398 pF
  - 4 de 1,498 pF
  - 12 de 680 pF
  - 4 de 68 pF
  - 4 de 1uF
  - 8 de 100 uF
- Diodos

- 5 diodos 1N4001

- Amplificadores operacionais
  - 6 TL082C
- Relés
  - 6 relés SH1NAC2
- Barra de pinos
  - Barra de pinos machos e fêmeas
- Driver de corrente
  - 1 ULN2001D

5.2 EQUIPAMENTOS UTILIZADOS

Para o desenvolvimento do projeto foram utilizados os seguintes equipamentos:

Figura 43 - Gerador de Funções ATF20B

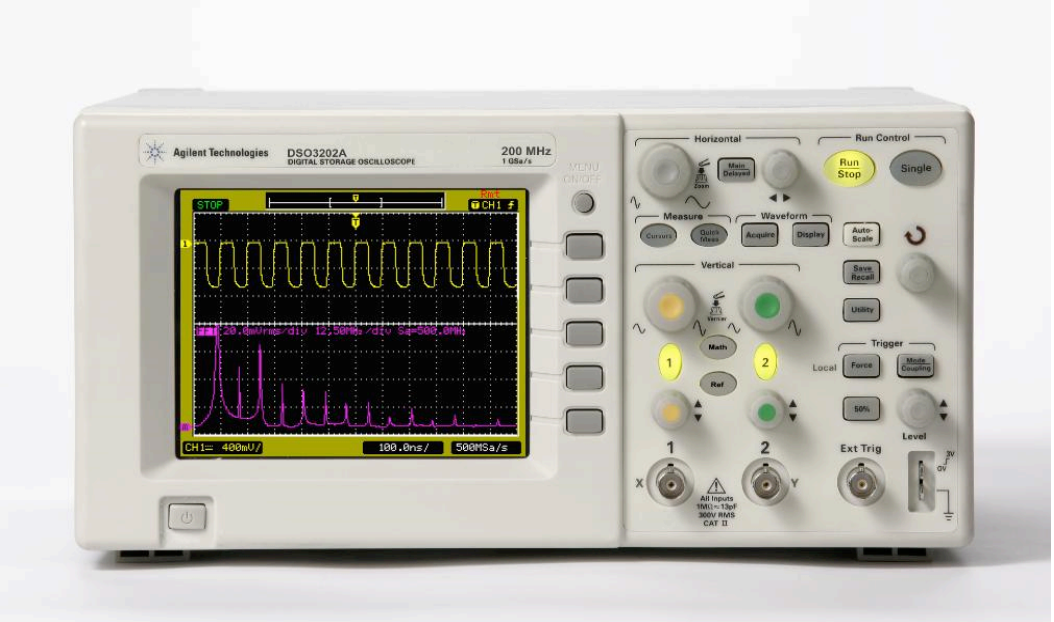


Figura 44 - Osciloscópio DSO3062A

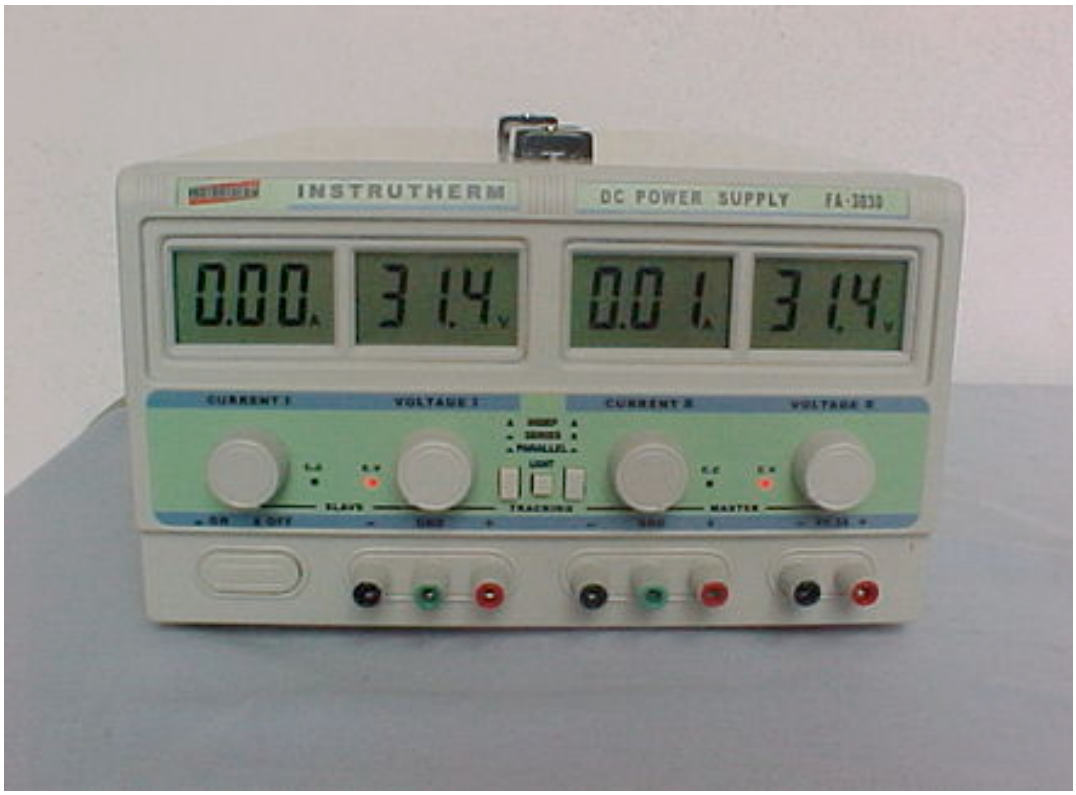


Figura 45 - Fonte de Alimentação do Departamento de Eletrônica da UTFPR

5.3 SISTEMA EMBARCARCADO

Nesta seção será explicado a implementação do projeto do sistema embarcado com trechos de código e detalhamento sobre o funcionamento deste.

O código é dividido em três partes, a primeira parte é onde ocorrem às inicializações de todos os *Cloks* do sistema e os periféricos são configurados para operarem no modo desejado. As funções Set_System e USB_Init realizam estas inicializações. Esta etapa de configuração será melhor explicada nos tópicos de configuração dos periféricos (5.2).

Após isso o programa entra em laço infinito em seu *main* como mostra a Figura 46. Neste laço são executadas os seguintes passos:

- Verifica no buffer de entrada do USB se chegou algum pacote.
- Espera o ADC estar inativo (Semáforo).
- Envia Frame de amostras para a estação base.
- Reativa ADC.

```
int main(void)
{
  Set_System();
  USB_Init();

  while (1)
  {
    verificaPacote();

    while( (ADC1->CR2 & ((uint32_t)0x00000001)) == 1 );// enquanto ADC não parar...

    delay();

    enviaPacoteDeDados();

    startADC();
  }
}
```

Figura 46 - Código main do software do microcontrolador

Como foi mostrado o código fica parado esperando o ADC parar de converter para seguir em frente a cada laço do programa. O único do programa sair desta espera é quando ocorre uma interrupção do DMA, que controla os dados que o ADC converte. Esta interrupção ocorre toda vez

que o DMA preenche um buffer de amostras do ADC no espaço de memória especificado em suas configurações, entrando então na terceira parte do código.

A terceira parte do programa é o tratador da interrupção do periférico DMA, que é mostrado na Figura 47. As etapas desse código são:

- Para o ADC, avisando o main que ele pode continuar sua execução
- Faz o complemento da flag “troca” (sinalizadora de qual espaço da memória deve ser utilizado pelo DMA).
- Verifica a flag “troca” para saber qual o valor deve ser colocado no registrador CMAR, que define o endereço base da memória que o DMA coloca os dados do ADC.
- Limpa a flag de interrupção do DMA.

```
void DMA1_Channel1_IRQHandler(void){

    stopADC();
    troca = troca?0:1; //complemento
    if ( troca == 1 ){
        DMA1_Channel1->CMAR = (uint32_t)&ADC1_ConvertedValueX2;
        //ADC_ConvertedValueX = ADC1_ConvertedValueX;
    }
    else{
        DMA1_Channel1->CMAR = (uint32_t)&ADC1_ConvertedValueX;
        //ADC_ConvertedValueX = ADC1_ConvertedValueX2;
    }

    DMA_ClearFlag(DMA1_FLAG_TC1);
}
```

Figura 47 - DMA Handler

Observe que esta troca de *buffer* é uma implementação de *dual buffer*. A *flag* “troca” é utilizada também quando o programa envia amostras do buffer para o USB, para este saber qual espaço da memória deve acessar, não causando conflitos de mexer no buffer que não foi preenchido.

5.3.1 Configuração dos periféricos

Este tópico descreve a configuração dos periféricos do microcontrolador.

5.3.1.1 Clock do core e dos periféricos

Os clocks de todos o sistema e dos periféricos é defino em cascata. Inicialmente definimos o clock do core e apartir deste são utilizados blocos de *Pré-scalers* para dividir este clock de referência por um número especificado nos registradores de configuração de cloks dos periféricos. A figura a3 mostra o código com as configurações mencionadas.

- HCLK (clock principal) 72Mhz.
- Barramento PCLK2 configurado em 72 MHz.
- Barramento PCLK1 configurado em 36 MHz.
- Clock do ADC configurado em 12 MHz.
- Bloco multiplicador de clock (PLL) configurado para multiplicar o cristal de 8 MHz por 9, gerando o clock de 72 MHz.
- Funções PeriphClockCmd abilitam os clocks dos periféricos específicados.
- Após configuração dos cloks, chama as funções de configurações dos periféricos.

```
* Description    : Configures Main system clocks & power
* Input          : None.
* Return         : None.
*******************************************************************************/
void Set_System(void)
{
  RCC_DeInit();
  RCC_HSEConfig(RCC_HSE_ON);
  HSEStartUpStatus = RCC_WaitForHSEStartUp();
  if (HSEStartUpStatus == SUCCESS)
  {
    FLASH_PrefetchBufferCmd(FLASH_PrefetchBuffer_Enable);
    FLASH_SetLatency(FLASH_Latency_2);
    RCC_HCLKConfig(RCC_SYSCLK_Div1);
    RCC_PCLK2Config(RCC_HCLK_Div1);
    RCC_PCLK1Config(RCC_HCLK_Div2);
    RCC_ADCCLKConfig(RCC_PCLK2_Div6);
    RCC_PLLConfig(RCC_PLLSource_HSE_Div1, RCC_PLLMul_9);
    RCC_PLLCmd(ENABLE);
    while (RCC_GetFlagStatus(RCC_FLAG_PLLRDY) == RESET){}
    RCC_SYSCLKConfig(RCC_SYSCLKSource_PLLCLK);
    while(RCC_GetSYSCLKSource() != 0x08){}
  }
  RCC_USBCLKConfig(RCC_USBCLKSource_PLLCLK_1Div5);
  RCC_APB1PeriphClockCmd(RCC_APB1Periph_USB, ENABLE);
  RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIO_DISCONNECT |
                         RCC_APB2Periph_GPIO_IOAIN , ENABLE);
  RCC_AHBPeriphClockCmd(RCC_AHBPeriph_DMA1, ENABLE);
  RCC_APB2PeriphClockCmd(RCC_APB2Periph_ADC1 | RCC_APB2Periph_GPIOC | RCC_APB2Periph_GPIOB | RCC_APB2Periph_GPIOA, ENABLE)
  RCC_APB2PeriphClockCmd( RCC_APB2Periph_TIM1, ENABLE);
  RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM2, ENABLE);
  RCC_AHBPeriphClockCmd(RCC_AHBPeriph_DMA1, ENABLE);

  Interrupts_Config();
  GPIO_Configuration();
  ADC_Configuration();
  TIMER_Configuration();
}
```

Figura 48 - Configurações dos clocks

Na habilitação dos clocks dos periféricos é necessário consultar o datasheet[11] para ver em qual barramento este está ligado.

5.3.1.2 GPIO

Os General Purpose Input/Output (GPIO), são periféricos para alterar o modo de operação dos pinos de entrada e saída do micro controlador, existem várias opções de configurações que podem ser consultadas no datasheet[11] do microcontrolador. Serão abordadas aqui apenas as configurações utilizadas no projeto. A Figura 49 e a Figura 50 mostram as configurações de todos os pinos utilizados.

- GPIO_MODE_AIN é o modo de operação "Entrada Analógica".
- Os 4 Canais do ADC são configurados no modo AIN.

```
/* Configure USB pull-up pin */
GPIO_InitStructure.GPIO_Pin = USB_DISCONNECT_PIN;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_OD;
GPIO_Init(USB_DISCONNECT, &GPIO_InitStructure);

GPIO_InitStructure.GPIO_Pin = GPIO_IOAIN_PIN;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AIN;
GPIO_Init(GPIO_IOAIN, &GPIO_InitStructure);

GPIO_InitStructure.GPIO_Pin = ADC_0;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AIN;
GPIO_Init(GPIO_IOAIN, &GPIO_InitStructure);

//ADC-1
GPIO_InitStructure.GPIO_Pin = ADC_1;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AIN;
GPIO_Init(GPIO_IOAIN, &GPIO_InitStructure);

//ADC-2
GPIO_InitStructure.GPIO_Pin = ADC_2;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AIN;
GPIO_Init(GPIO_IOAIN, &GPIO_InitStructure);

//ADC-3
GPIO_InitStructure.GPIO_Pin = ADC_3;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AIN;
GPIO_Init(GPIO_IOAIN, &GPIO_InitStructure);
```

Figura 49 - Configuração GPIO

- Pinos de controle dos relés são colocados em modo push-pull de saída.
- O pino A1 é colocado no modo Função alternante para saída de PWM para o controle de offset do sistema de condicionamento de sinal.

```
//Configuração do pinos de controle dos relés
GPIO_InitStructure.GPIO_Pin = PC8;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;//
GPIO_Init(GPIOC_OUT, &GPIO_InitStructure);

GPIO_InitStructure.GPIO_Pin = PB8;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;//
GPIO_Init(GPIOB_OUT, &GPIO_InitStructure);

GPIO_InitStructure.GPIO_Pin = PB9;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;//
GPIO_Init(GPIOB_OUT, &GPIO_InitStructure);

GPIO_InitStructure.GPIO_Pin = PC6;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;//
GPIO_Init(GPIOC_OUT, &GPIO_InitStructure);

GPIO_InitStructure.GPIO_Pin = PC7;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;//
GPIO_Init(GPIOC_OUT, &GPIO_InitStructure);

GPIO_InitStructure.GPIO_Pin = PC10;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;//
GPIO_Init(GPIOC_OUT, &GPIO_InitStructure);
/*I/O (digital, controle de relés)
    PWM
'A8 pwm0 ext1-2 pino 3

/*GPIOA Configuration: TIM3 channel 1 and 2 as alternate function push-pull */
GPIO_InitStructure.GPIO_Pin =  GPIO_Pin_1;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;

GPIO_Init(GPIOA, &GPIO_InitStructure);
```

Figura 50 - Configuração GPIO segunda parte

Observe que cada canal de cada Timer está relacionado com apenas um pino de saída do microcontrolador. Neste caso o canal 1 do Timer 2 está ativo. Para informações sobre a relação dos Timers com os pinos consultar a tabela de funções dos pinos no datasheet[11] do microcontrolador.

5.3.1.3 ADC

O Conversor Analógico Digital foi configura de modo a atender as exigências do projeto. Essas configurações são as lista das abaixo e mostradas no código da figura a6.

- Configurado em modo independente.
- Desabilita opção de escaneamento dos canais do ADC.
- Habilita opção de conversão continua de dados.
- Sem *trigger* externo.
- Alinha os 12bits de dados à direita do registrador de saída do ADC.
- Seleciona canal do ADC e configura como canal regular a coloca o tempo de amostragem no máximo.
- Ativa o ADC.
- Realiza etapa de calibração do ADC.

```
void ADC_Configuration(void)
{
  ADC_InitTypeDef ADC_InitStructure;

  /* ADC1 configuration ------------------------------------------------------*/
  ADC_InitStructure.ADC_Mode = ADC_Mode_Independent;
  ADC_InitStructure.ADC_ScanConvMode = DISABLE;
  ADC_InitStructure.ADC_ContinuousConvMode = ENABLE;
  ADC_InitStructure.ADC_ExternalTrigConv = ADC_ExternalTrigConv_None;
  ADC_InitStructure.ADC_DataAlign = ADC_DataAlign_Right;
  ADC_InitStructure.ADC_NbrOfChannel = 1;
  ADC_Init(ADC1, &ADC_InitStructure);

  /* ADC1 regular channel configuration */
  ADC_RegularChannelConfig(ADC1, ADC_AIN_CHANNEL, 1, ADC_SampleTime_1Cycles5);

  /* Enable ADC1 DMA */
  ADC_DMACmd(ADC1, ENABLE);

  /* Enable ADC1 */
  ADC_Cmd(ADC1, ENABLE);

  /* Enable ADC1 reset calibaration register */
  ADC_ResetCalibration(ADC1);
  /* Check the end of ADC1 reset calibration register */
  while(ADC_GetResetCalibrationStatus(ADC1));

  /* Start ADC1 calibaration */
  ADC_StartCalibration(ADC1);
/* Check the end of ADC1 calibration */
  while(ADC_GetCalibrationStatus(ADC1));
}
```

Figura 51 - Configuração ADC

5.3.1.4 DMA

O DMA foi utilizado para transportar os dados convertido pelo ADC para a memória sem a necessidade de sobrecarregar o microprocessador com está tarefa. Apenas avisando este quando um número suficiente (um buffer) de amostras for armazenado. Conseguindo assim atingir a máxima performance do sistema de conversão. O código com as configurações é mostrado na Figura 52.

- Passa o “endereço” do ADC1 para o DMA.
- Passa o endereço da memória em que os dados serão armazenados.
- Atribui o tamanho do buffer.

- Desabilita opção de mudar de periférico incrementando o endereço.
- Habilita opção de incremento de memória para percorrer o buffer inteiro da memória
- Utiliza meia palavra (16bits).
- Ativa modo circular para voltar para o endereço base de memória quando este chegar ao fim do buffer.
- Coloca prioridade alta no DMA
- Habilita DMA.
- Configura Interrupção do DMA para acontecer quando o buffer inteiro for preenchido.

```
DMA_InitTypeDef DMA_InitStructure;
/* DMA1 channel1 configuration ----------------------------------------------*/
DMA_DeInit(DMA1_Channel1);
DMA_InitStructure.DMA_PeripheralBaseAddr = ADC1_DR_Address;
DMA_InitStructure.DMA_MemoryBaseAddr = (uint32_t)&ADC1_ConvertedValueX;
DMA_InitStructure.DMA_DIR = DMA_DIR_PeripheralSRC;
DMA_InitStructure.DMA_BufferSize = 1024;
DMA_InitStructure.DMA_PeripheralInc = DMA_PeripheralInc_Disable;
DMA_InitStructure.DMA_MemoryInc = DMA_MemoryInc_Enable;
DMA_InitStructure.DMA_PeripheralDataSize = DMA_PeripheralDataSize_HalfWord;
DMA_InitStructure.DMA_MemoryDataSize = DMA_MemoryDataSize_HalfWord;
DMA_InitStructure.DMA_Mode = DMA_Mode_Circular;
DMA_InitStructure.DMA_Priority = DMA_Priority_High;
DMA_InitStructure.DMA_M2M = DMA_M2M_Disable;
DMA_Init(DMA1_Channel1, &DMA_InitStructure);

/* Enable DMA1 channel1 */
DMA_Cmd(DMA1_Channel1, ENABLE);

/* Enable the DMA1 Channel1 Transfer complete interrupt */
DMA_ITConfig(DMA1_Channel1, DMA_IT_TC, ENABLE);
```

Figura 52 - Configuração DMA

5.3.1.5 Timer

O canal 1 do timer 2 foi configurado para gerar um PWM de saida que é utilizado no offset do sinal no condicionamento do sinal. Esta configuração é mostrada na Figura 53.

```
void TIMER_Configuration(){

  TIM_TimeBaseInitTypeDef  TIM_TimeBaseStructure;
  TIM_OCInitTypeDef  TIM_OCInitStructure;
  u16 CCR2_Val = 375;
  RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM2, ENABLE);

  TIM_TimeBaseStructure.TIM_Period = 0x05C9;
  TIM_TimeBaseStructure.TIM_Prescaler = 0;
  TIM_TimeBaseStructure.TIM_ClockDivision = 0;
  TIM_TimeBaseStructure.TIM_CounterMode = TIM_CounterMode_Up;

  TIM_TimeBaseInit(TIM2, &TIM_TimeBaseStructure);

  /* PWM1 Mode configuration: Channel1 */
  TIM_OCInitStructure.TIM_OCMode = TIM_OCMode_PWM1;
  TIM_OCInitStructure.TIM_OutputState = TIM_OutputState_Enable;
  TIM_OCInitStructure.TIM_Pulse = CCR2_Val;
  TIM_OCInitStructure.TIM_OCPolarity = TIM_OCPolarity_High;

  TIM_OC2Init(TIM2, &TIM_OCInitStructure);

  TIM_OC2PreloadConfig(TIM2, TIM_OCPreload_Enable);

  TIM_ARRPreloadConfig(TIM2, ENABLE);

  /* TIM3 enable counter */
  TIM_Cmd(TIM2, ENABLE);
}
```

Figura 53 - Configuração timer

O Timer é configurado como *output counter* (OC) no modo PWM1 [11].

5.3.1.6 Interrupções

As Interrupções utilizadas neste projeto foram a do USB e do DMA. Como mostras o código da Figura 54.

```
void Interrupts_Config(void)
{
  NVIC_InitTypeDef NVIC_InitStructure;

  NVIC_PriorityGroupConfig(NVIC_PriorityGroup_2);

  NVIC_InitStructure.NVIC_IRQChannel = USB_LP_CAN1_RX0_IRQn;
  NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 1;
  NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0;
  NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
  NVIC_Init(&NVIC_InitStructure);

  /* Enable the DMA1 Channel1 Interrupt */
  NVIC_InitStructure.NVIC_IRQChannel = DMA1_Channel1_IRQn;
  NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 2;
  NVIC_Init(&NVIC_InitStructure);

}
```

Figura 54 - Configuração Interrupções

5.3.2 Calibragem da taxa de amostragem

O sistema do osciloscópio possui sete configurações de base de tempo, esta são: 5 us/div, 50 us/div, 0.5 ms/div, 5 ms/div, 50 ms/div, 0.5 s/div e 1 s/div, cada uma se adequa a um certo alcance de frequência de maneira que todas as frequencias que estejam dentro da proposta possam ser medidas. Porém percebeu-se que a taxa de amostragem do ADC não precisa e não deve ser a máxima em todas as ocasiões, pois de nada adianta ter muitos pontos que não representam a onda. Passamos então adotar o conceito de pontos representativos, com isso ajustamos a taxa de amostragem de cada base de tempo para se adequar ao sinal e pegar amostras representativas deste. As taxas obtidas foram as seguintes:

- 5 us/div ➔ 1,2 MSps
- 50 us/div ➔ 439 KSps
- O,5 ms/div ➔ 71,428 KSps
- 5 ms/div ➔ 8928 KSps
- 50 ms/div ➔ 2232 KSps
- 0,5 s/div ➔ 120 Sps
- 1 s/div ➔ 60 Sps

5.3.3 Dificuldades encontradas

No sistema embarcado foi encontrada a seguinte dificuldade:

- Conseguir taxas de conversão de 1 MSps com a precisão requerida no projeto.

Este problema foi resolvido com um extenso estudo sobre o funcionamento do ADC e do DMA. Tentando aperfeiçoar ao máximo as rotinas onde a conversão de dados estava presente. Deixando o DMA tomar conta do ADC a maior parte do tempo da conversão e só utilizando o core na hora de trocar o buffer do DMA.

5.4 CONDICIONAMENTO DE SINAL

5.4.1 Calibragem do circuito de condicionamento de sinal

A calibragem do circuito consiste em encontrar valores de capacitores que compensam as capacitâncias intrínsecas ao circuito, vide 5.4.3. Cada PCI tem que ser calibrada meticulosamente observando quando a saída do circuito de aquisição de sinal está se comportando como o esperado.

5.4.2 Confecção da PCI

Para a confecção da PCI foi necessário criar os arquivos gerberer dos furos e das trilhas. Para isso também foi utilizado o software Altium Summer Designer. Os arquivos com a informação dos furos e o arquivo com a informação das trilhas são mostrados abaixo, sendo o primeiro o arquivo dos furos e o segundo das trilhas.

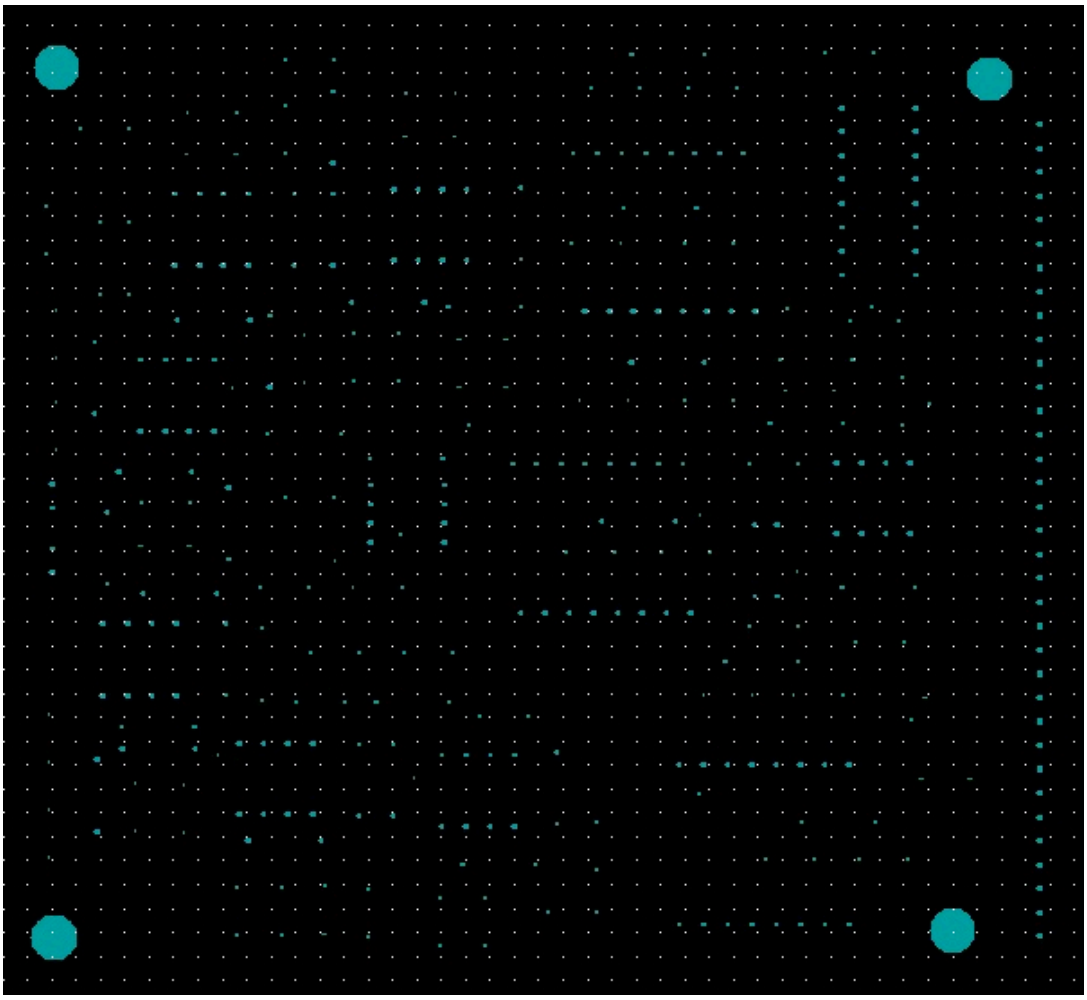


Figura 55 - Gerberer de furos do SAD

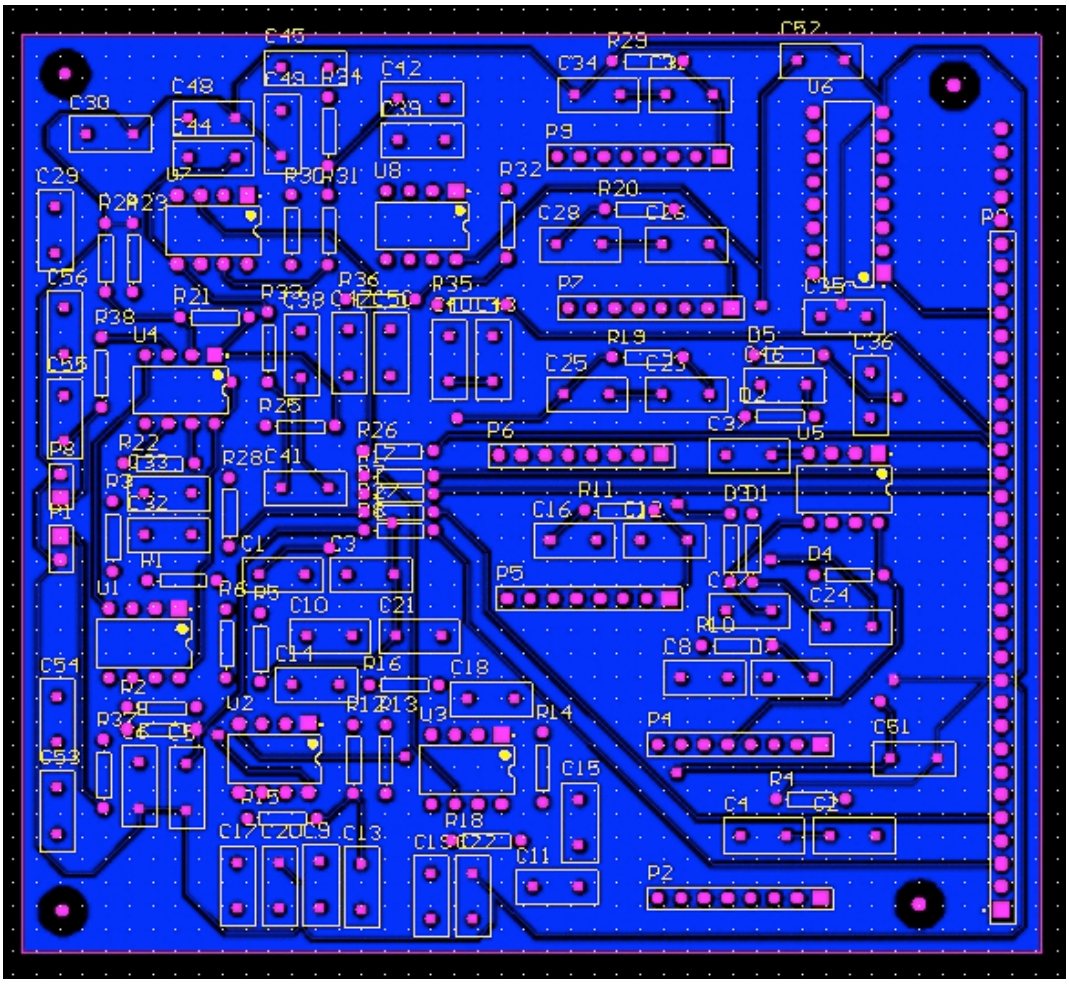


Figura 56 - Gerberer de trilhas do SAD

Após a geração de todos os arquivos necessários, a confecção da placa foi encomendada ao professor Fernando Cardoso Castaldo. A qualidade da placa produzida é satisfatória se for levado em conta o custo da mesma.

5.4.3 Dificuldades encontradas

Primeiramente a equipe se viu em um impasse sobre as atenuações e amplificações feitas sobre o sinal de entrada. Não sabíamos ate então qual o numero e os valores de atenuação e amplificação que o sistema de aquisição de dados deveria possuir. Após o circuito ter sido reformulado algumas vezes chegamos à conclusão que três níveis de atenuação e um valor fixo de amplificação seriam o suficiente para se ter o melhor nível de resolução e o maior nível de segurança, tendo a premissa de que o usuário

vai injetar no sistema de aquisição de dados um sinal de no máximo 20 Volts pico a pico.

Após esta dificuldade ter sido superada a equipe montou o circuito projetado, porém o circuito era puramente resistivo, pois naquele momento a equipe achou que o uso de capacitores não se fazia necessário. Mas os resultados práticos mostraram que o uso de capacitores se fazia necessário, pois sem eles existiam capacitâncias intrínsecas no circuito que estavam deformando a forma e o valor de tensão do sinal do entrada fazendo com que o sinal na saída não fosse como o esperado pela equipe. O resultado disto foi que a equipe necessitou elaborar um novo circuito e nesse circuito optou-se por colocar capacitores em paralelo com as resistências dos divisores resistivos responsáveis por fazer a atenuação do sinal de entrada. Para resolvermos o problema os capacitores foram escolhidos de modo que eles agissem como filtro corta faixa, sendo que a frequência de corte inferior e superior fossem iguais compensando assim as capacitâncias intrínsecas do sistema. Abaixo segue figuras representando o divisor resistivo, o divisor resistivo com a capacitância intrínseca e o divisor elaborado par o projeto, consecutivamente. Para que a compensação seja feita corretamente deve-se aplicar a formula:

$$R1*C1 = R2*(C2+ C\ intrínseco)$$

Esta formula pode ser deduzida escrevendo a função de transferência do circuito. Quando essa restrição é satisfeita a função de transferência é reduzida à:

$$Vout = Vin * (R1 / (R1+ R2)$$

Demonstrando que o comportamento desejado do sistema foi alcançado.

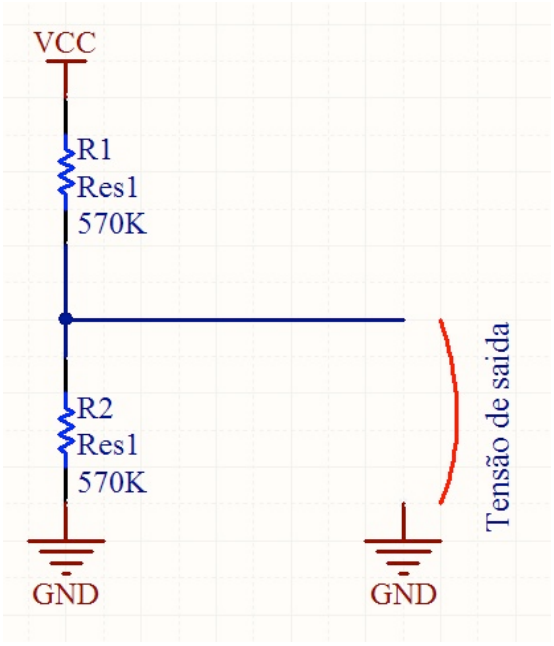


Figura 57 - Divisor puramente resistivo

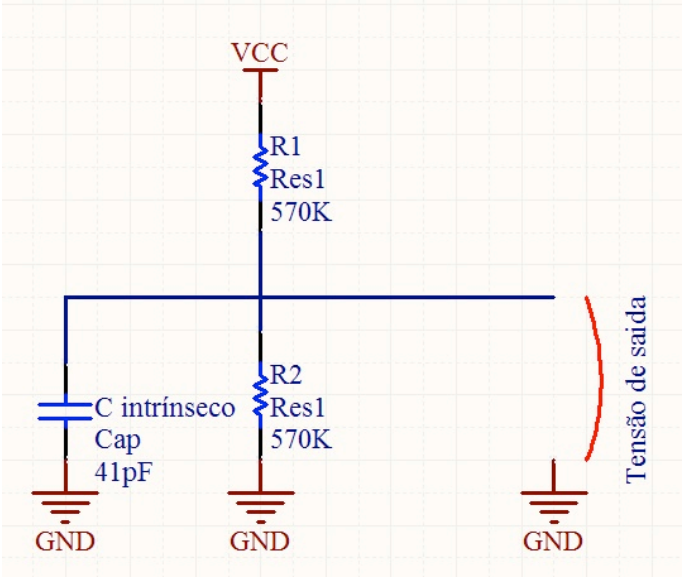


Figura 58 - Divisor com capacitância intrínseca

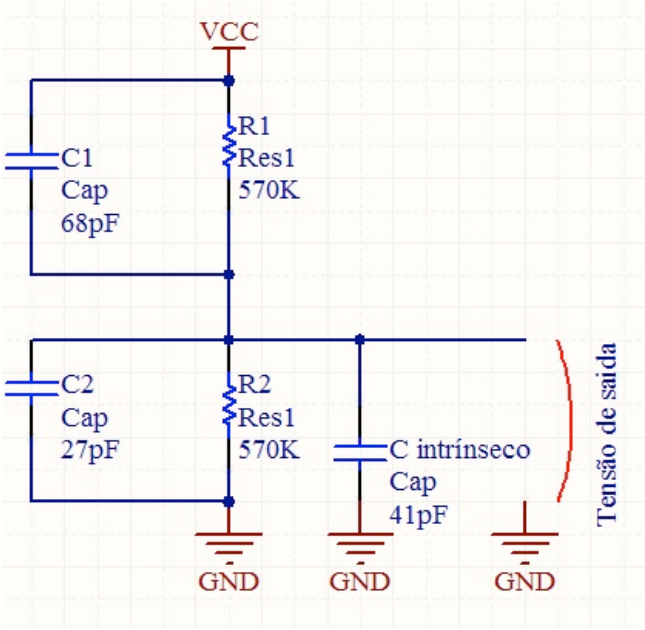


Figura 59 - Divisor resistivo elaborado para o projeto

$$R1 * C1 = R2 * (C2 + CIntrinseca)$$

Figura 60 – Formula para o calculo da compensação

Outra dificuldade enfrentada pela equipe possui ligação com a alimentação dos amplificadores operacionais que fazem parte do sistema de aquisição de dados. A equipe desejava achar uma solução que não restringisse a portabilidade do produto final. Logo procuramos um modo de fazer com que a alimentação de todo o sistema fosse provida pela própria porta USB. Desse modo o usuário não precisaria ficar carregando baterias ou fontes juntamente com o osciloscópio. A solução encontrada foi utilizar um charge pump. Este componente transformava a tensão de +5V em +7V e -7V . Com isso conseguimos fornecer a alimentação para todo o sistema de aquisição de sinal.

Porém a alimentação fornecida pelo charge pump se mostrou muito ruidosa e para sanar esse problema foi necessário a de uma fonte externa. Porém uma melhor solução seria a utilização de um regulador de tensão na saída do charge pump.

5.5 COMUNICAÇÃO

5.5.1 Implementação do sistema embarcado

A implementação do protocolo no sistema embarcado é mais simples do que na estação base, pois este se comporta apenas como escrava na comunicação.

Basicamente o software faz uma verificação periódica (*polling)* em seu buffer de entrada. Quando Há pacote ele verifica este pacote, se estiver correto trata este pacote, acerta as configurações do microcontrolador e responde este pacote.

Sempre que possível o microcontrolador está enviando frames de amostras para a estação base, sem esperar resposta desta.

5.5.2 Implementação da estação base

O protocolo implementado na estação-base possui sua própria classe (ProtocoloComunicacao), implementa a classe runnable, e possui relação de associação com a classe Comunicação visto que escreve e lê dados no buffer de comunicação.

O protocolo foi feito baseado na máquina de estados projetada na seção 4.3.1.1. O método run() verifica em qual estado o protocolo está através de um switch, executando o estado (método) em questão. Quando um estado é executado e, em alguns casos, determinadas condições atendidas ele passa para o próximo estado.

Os dados recebidos da comunicação são sempre armazenados em uma palavra de dois bytes, e somente tem efeito no algoritmo caso essa palavra esteja completa. O modo mais comum do protocolo de comunicação é de receber amostras através da comunicação, porém quando um comando é enviado para o sistema embarcado, ele somente armazena amostras no buffer do canal quando o ack do comando é recebido, com isso o usuário não recebe informações desatualizadas no plotador.

O protocolo também controla a escrita de pacotes na comunicação, quando um usuário seleciona, por exemplo, o ant-aliasing, uma flag

(warnEmb situada na classe controle) informa que a estação base deve enviar um pacote informando ao microcontrolador as novas configurações requisitadas. Porém esse pacote só será enviado caso o estado atual do protocolo seja o esperandoEnvioDeComandos.

5.5.3 Dificuldades encontradas

- Biblioteca USB em Java na estação-base

A equipe havia planejado utilizar conexão USB inicialmente no projeto, a biblioteca que utilizaríamos como mostra no planejamento é a jUSB, porém a classe disponível no firmware do sistema embarcado (HID - Human Interface Device) não correspondia à dessa biblioteca, se quiséssemos utilizá-la teríamos que desenvolver um driver para isso tornando-se inviável. A equipe mudou para outra biblioteca USB a libUSB que para Java já havia um wrapper pois é nativa do Windows, escolhemos essa biblioteca por possuir driver para a classe USB HID, e conseguimos fazer a conexão do sistema embarcado com a estação base, porém está apresentou-se muito lerda, apesar de verificarmos as configurações da conexão não identificamos o problema, em geral demorava cerca de 3 segundos para recebermos um pacote enviado do sistema embarcado, inviabilizando essa outra biblioteca.

Então a equipe decidiu mudar a abordagem e utilizar comunicação serial, apesar de ter taxas inferiores de velocidade em comparação com a conexão USB, isso não inviabilizaria o projeto por dois motivos:

- Quando o usuário seleciona uma base de tempo alta (1s/div), o microcontrolador diminui a taxa de amostragem do ADC, pois não precisamos de um milhão de amostras por segundo, pois o próprio plotador possui a precisão de pixeis e descartaria pontos plotados na mesma posição do pixel. Então podemos abaixar a taxa de amostragem do ADC para uma freqüência menor atendendo a velocidade da conexão.

  Uma desvantagem de utilizar essa abordagem é o aparecimento de aliasing-visual, porém até os osciloscópios profissionais não escapam desse fenômeno

- Quando o usuário seleciona uma base de tempo baixa (5us/div) o microcontrolador precisa estar na máxima taxa de amostragem, porém ele pode enviar um frame de amostras que complete a janela do plotador na estação base, esperar essas amostras serem enviadas, e preparar outro frame depois de um determinado tempo. A espera para enviar outro frame de amostras faz o “sincronismo” da comunicação, e o buffer de entrada da comunicação na estação-base não é lotado.

Outro problema encontrado foi a falta de controle sobre o buffer de saída. Como este buffer é circular, ou seja, quando ultimo dado do buffer é enviado o ponteiro para próximo dado a ser enviado volta para o inicio do buffer, este gera problemas de descontinuidade no sinal plotado no software da estação base.

5.6 ESTAÇÃO BASE

5.6.1 Método de plotagem

O software possui dois modos de desenhar as amostras que são recebidas através da comunicação. Esses dois modos são selecionados pela base de tempo que o usuário escolhe, e a principal diferença entre essas abordagens é a forma que o plotador é atualizado. O primeiro modo desenha as amostras no plotador, quando este está ocioso, no momento em que elas são recebidas, este modo foi criado para o usuário poder observar o desenho da onda o mais rápido possível (em ondas de baixas frequências por exemplo).

No segundo modo o plotador é atualizado somente quando completa um frame de amostras, ou seja quando a tela é totalmente preenchida, esse caso dá privilégio à ondas de alta frequência visto que diferentemente do modo um, o interessante para o usuário é analisar um frame inteiro de amostras, pois a base de tempo selecionada geralmente é muito baixa e fica impossível o acompanhamento de um ser humano da formação de onda. Para o modo dois também foi utilizado um triple buffer, que somente

desenha na tela um frame após este ser completado, enquanto isso amostras que chegam da comunicação são armazenadas em outro buffer, outra característica é que o triple buffer implementado é circular.

Em nosso software as bases de tempo que correspondem ao modo um são: 1s/div, 0.5s/div, e ao modo dois: 50ms/div, 5ms/div, 0.5ms/div, 50us/div, 5us/div.

5.6.2 Medições

5.6.2.1 frequência

O método utilizado para a medição de frequências foi baseado no zero-crossing do sinal em relação à um determinado eixo horizontal. O eixo horizontal utilizado para fazer essa medição foi o trigger. Ou seja, o usuário precisa ativar o trigger para que a frequência do sinal seja calculada.

O algoritmo funciona da seguinte forma: Quando ocorre três vezes o zero-crossing do sinal amostrado em relação à posição do trigger, o tempo relativo entre a primeira e terceira vez que o sinal passa pelo trigger é armazenado e o inverso desse valor corresponde a frequência.

5.6.2.2 Tensão pico a pico

A tensão pico à pico calculada utiliza métodos da classe XYSeries que retornam os valores máximo e mínimo de determinada série. O método implementado para o cálculo de tensão pico a pico retorna o módulo da diferença entre o valor máximo e mínimo da série.

5.6.2.3 Tensão Rms

O cálculo da Tensão RMS utiliza a fórmula da Figura 61. A partir de uma série de pontos que representam valores de tensão obtemos o valor RMS, na fórmula x_i representa um valor de tensão e N o tamanho da série.

$$x_{\mathrm{rms}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2} = \sqrt{\frac{x_1^2 + x_2^2 + \cdots + x_N^2}{N}}$$

Figura 61 - Tensão RMS através de uma série de pontos

5.6.3 Cursores

A biblioteca JFreechart não possui nenhum tipo de cursor que atende a necessidade da equipe, mas possui marcadores. Marcadores são linhas que podem ser associados ao plotador, elas podem ser verticais ou horizontais, através desses marcadores simulamos em nosso software cursores verticais (o trigger foi implementado dessa forma também). Quando os cursores estão ativos, o frame do projeto gera um evento caso o usuário aperte um botão no plotador, e dependendo do cursor habilitado, ele transporta-o para onde o usuário apertou o botão.

Os cursores retornam dados relacionados ao gráfico plotado que representam valores de tensão e tempo, a tensão que é relacionada a um cursor é aquela em que o sinal cruza o eixo vertical do cursor, além disso, é possível analisar valores relativos de tensão e tempo entre cursores.

5.6.4 Trigger

O trigger implementado participa do controle do plotador informando quando ele deve começar a desenhar as amostras. Quando ativo permite que o plotador desenhe amostras somente quando essas cruzarem pela posição do trigger, ou seja, buscamos o zero-crossing em relação à posição do trigger. Além disso, foi utilizado também um schmitt trigger para fazer este controle, o nível de tensão usado para acionar o trigger foi de 0.1V (visto que o sistema alimentado com fonte de alimentação externa gera ruído na ordem de 0.02V)

5.6.5 Stop

O stop corresponde a uma variável de controle do plotador, que libera ou congela o gráfico desenhado. O valor dessa variável é alterado quando o usuário aperta o botão correspondente.

5.6.6 Simulação de sinais para a estação base

Devido ao fato do desenvolvimento do sistema de aquisição de dados ocorrer em paralelo com o desenvolvimento do software da estação base, foi necessário simular um gerador de funções dentro de nosso software gerando formatos de ondas com tensões e frequências diversas. O objetivo desta simulação foi testar o funcionamento do software na estação base incluindo os métodos de cálculo de frequência, tensão pico a pico e RMS.

Este simulador está incluso no projeto do software no pacote Testes.

5.6.7 Dificuldades encontradas

As principais dificuldades encontradas na implementação do software da estação base foram:

- Software ocupando muito processamento do PC

O software na estação-base ficou lento, o provável motivo desse problema foi o controle de acesso às threads que foi feito somente com flags. Uma possível solução para este problema seria a utilização de semáforos junto com flags para o controle das threads no programa.

- O modo de desenho de amostras em tempo real, não ocorrer realmente em tempo real.

O modo de desenho um do plotador explicado na seção 5.4.1 não pode ser efetivamente realizado. As amostras são atualizadas na tela em pequenos frames, portanto o usuário não observa a formação da onda amostra por amostra, além disso, o algoritmo implementado segue a seguinte rotina: Caso o plotador estiver livre e há novas amostras no buffer de dados recebidos plote essas novas amostras. O principal motivo de o problema existir é que apesar do software da estação-base possuir uma thread (Controle) em que uma das responsabilidades é a de liberar o plotador para que desenhe as amostras que chegam através da

comunicação, pode ocorrer que o próprio sistema operacional necessite do processador para alguma operação, além disso, o próprio sistema da estação-base possui ao todo quatro threads, que torna o aplicativo um tanto quanto “pesado”.

6 Resultados

6.1 INTERFACE GRÁFICA DO OSCILOSCÓPIO

A Figura 62 corresponde a interface gráfica funcionando corretamente com um canal. No exemplo já a conexão já está estabelecida e capturando um sinal senoidal a escala está em 4 Volts/divisão obtendo assim uma tensão RMS de 6,48 Volts, tensão pico à pico com 18,49 Volts e uma frequência de 9,97 kHz.

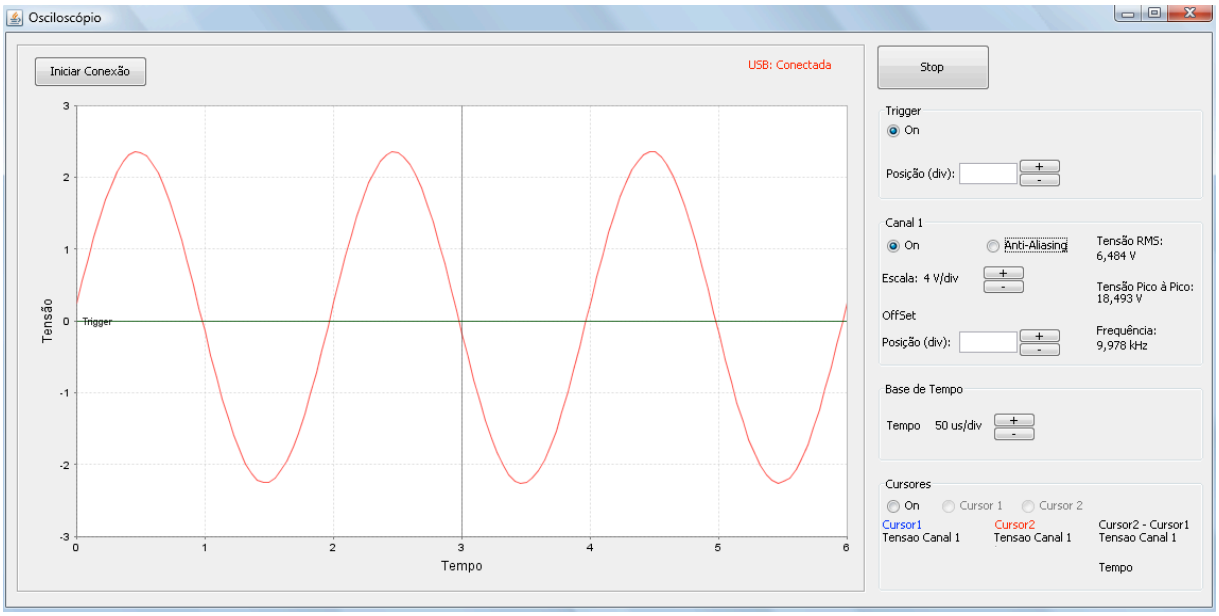


Figura 62 - Interface gráfica software

6.2 ANÁLISE DA PLACA DE CIRCUITO IMPRESSO

A placa de circuito impresso produzida teve um resultado satisfatório, porém poderia receber alguns melhoramentos como melhorar o lugar onde ficam as legendas, fazer com que algumas trilhas fiquem ainda menores, melhorar o posicionamento dos relés. Estes erros foram cometidos devido à inexperiência de roteamento e fabricação de uma placa de circuito impresso. É valido salientar que nenhum desses erros causou o não funcionamento da placa produzida.

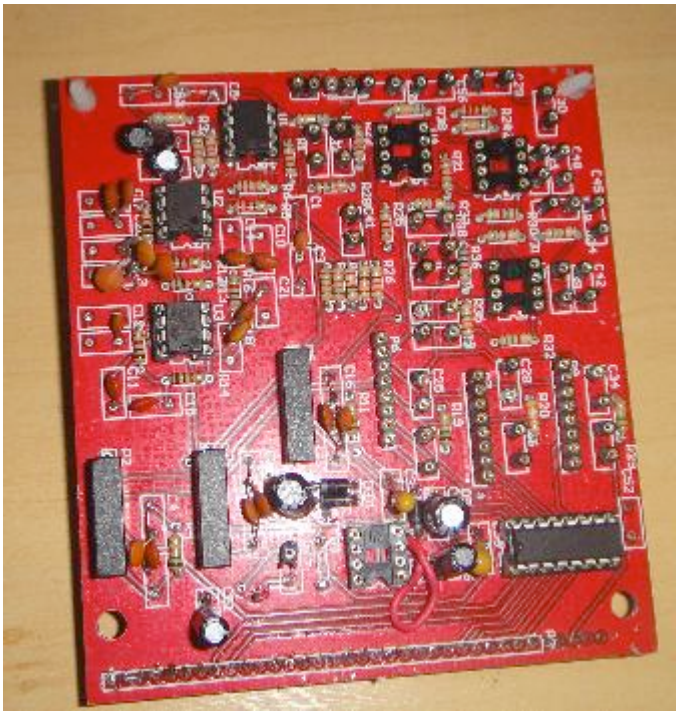


Figura 63 – PCI do sistema de aquisição de sinais

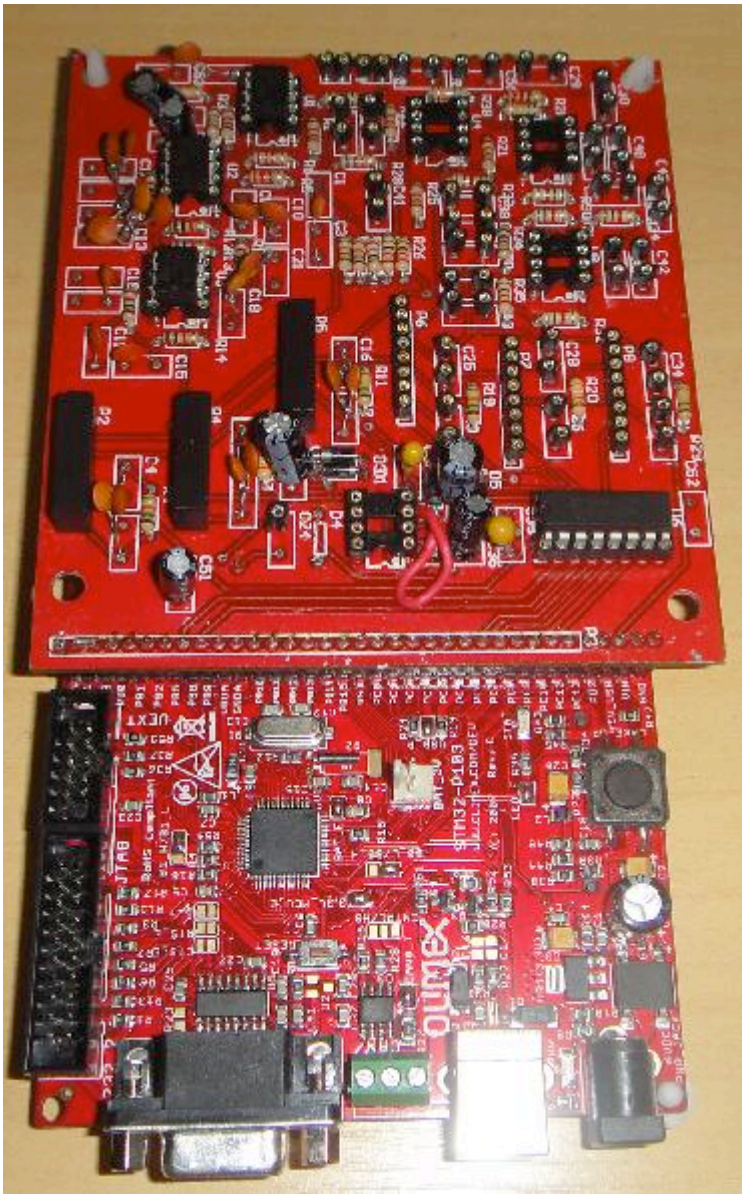


Figura 64 – PCI do sistema de aquisição de sinais acoplada ao kit de desenvolvimento

6.2.1 Comparação com Outros Osciloscópios

Neste tópico está sendo apresentado os resultados obtidos com o osciloscópio desenvolvido pela equipe, para isto foi produzido vários sinais de entrada com frequências, tensões pico a pico e tensões rms variados. Para se ter uma ideia de qual foi o desempenho do osciloscópio projeto em comparação com o osciloscópio comercial foi calculado a diferença entre os valores medidos por ambos. Após as imagens abaixo estão os resultados obtidos:

- Frequência 1Hz e Tensão Pico a Pico 20,2 V

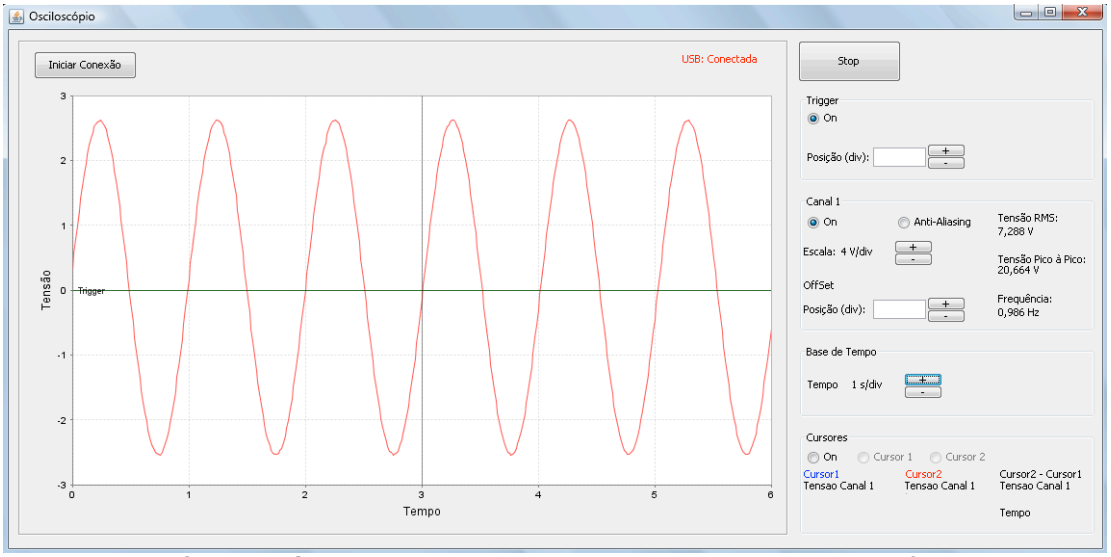


Figura 65 - Osciloscópio projetado [Vpp = 20,664 V; Vrms = 7,288 V; f = 0,986 Hz]

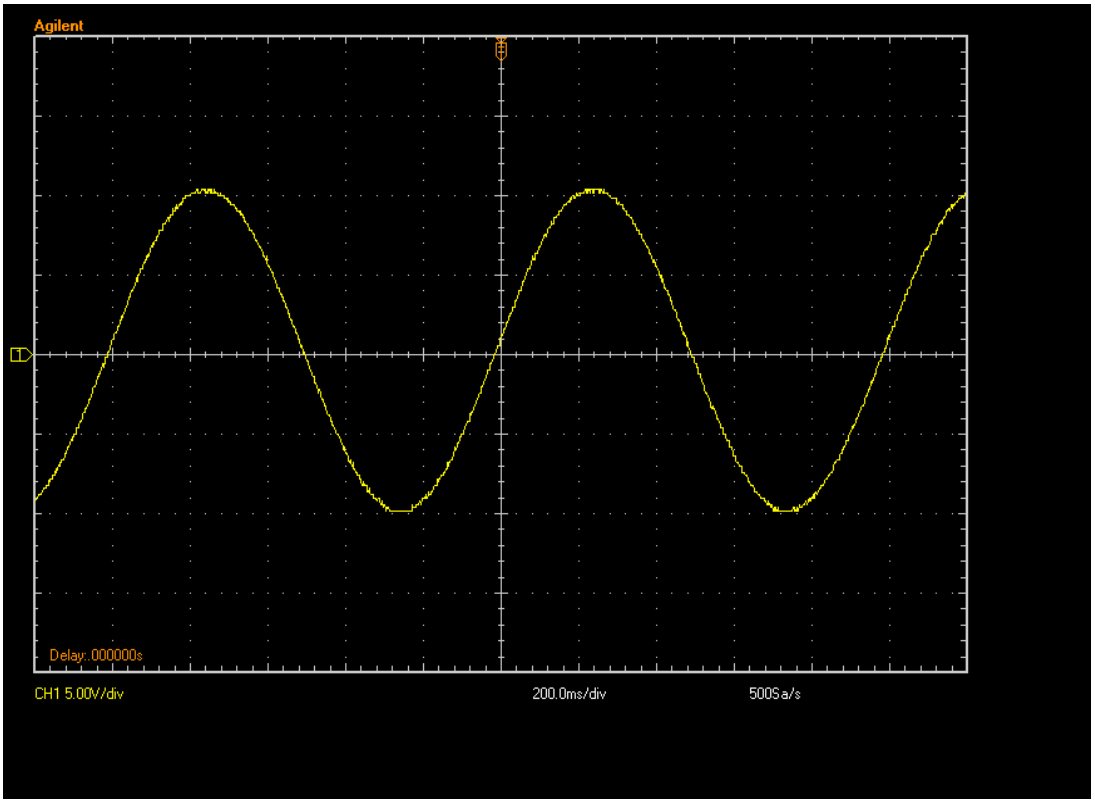


Figura 66 - Osciloscópio comercial [Vpp = 20,2 V; Vrms = 7 V; f = 1 Hz]

- Frequência 1 kHz e Tensão Pico a Pico 118 mV

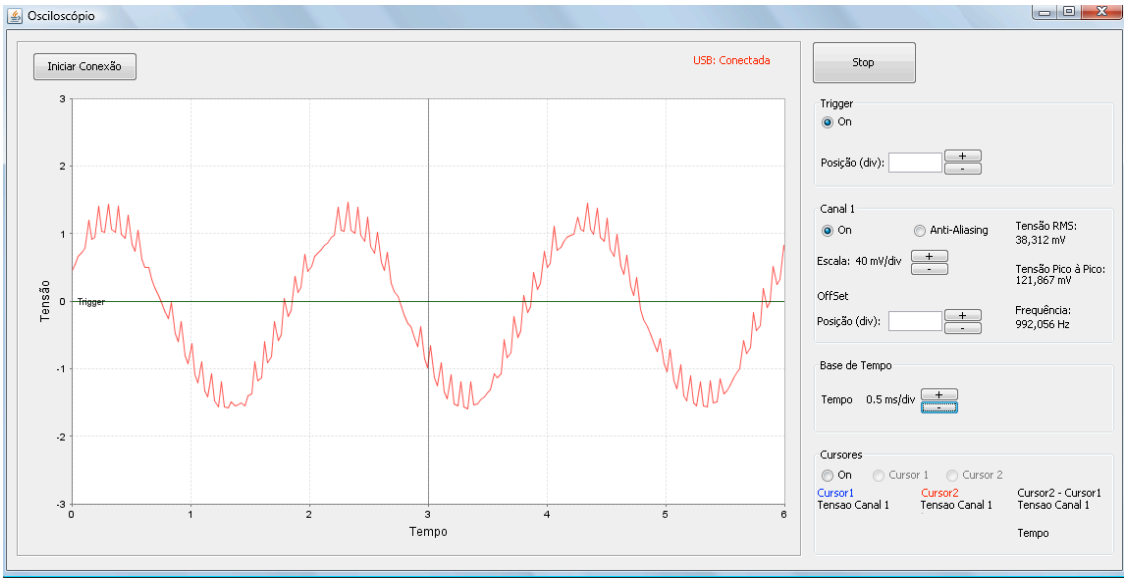


Figura 67 - Osciloscópio projetado [Vpp = 121,867 mV; Vrms = 38,312 mV; f = 992,056 Hz]

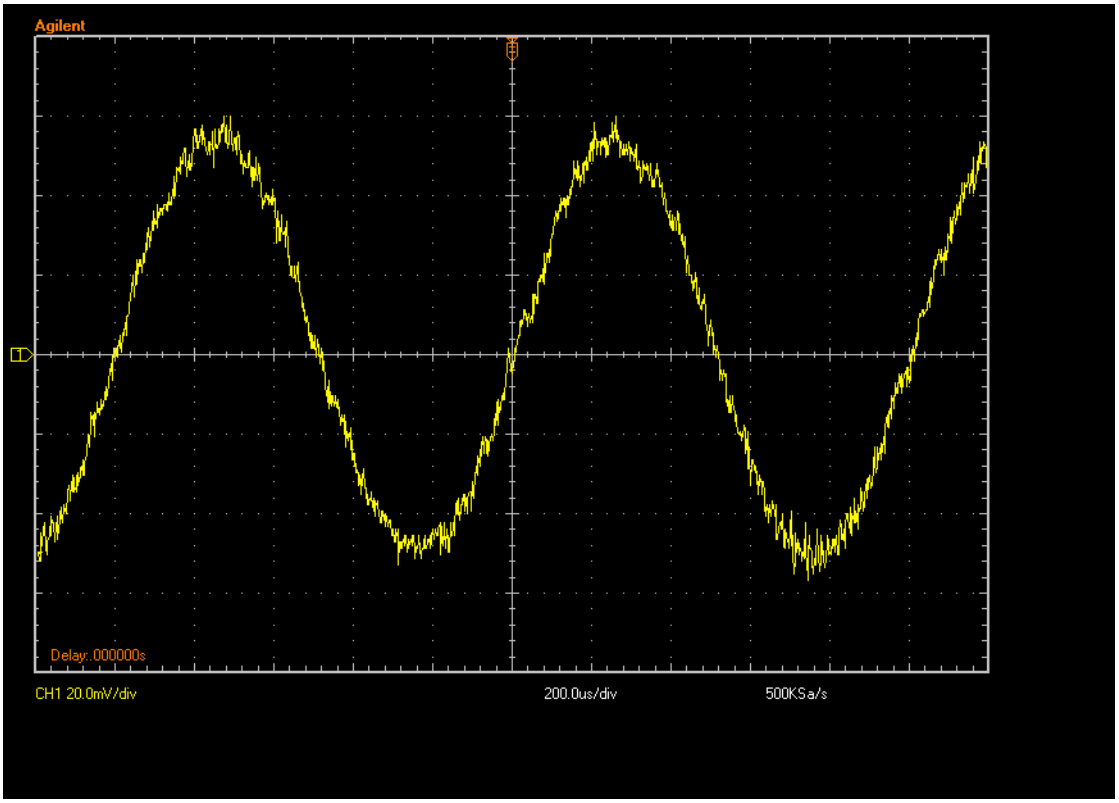


Figura 68 - Osciloscópio comercial [Vpp = 118 mV; Vrms = 36,5 mV; f = 1 kHz]

- Frequência 1 kHz e Tensão Pico a Pico 123 mV

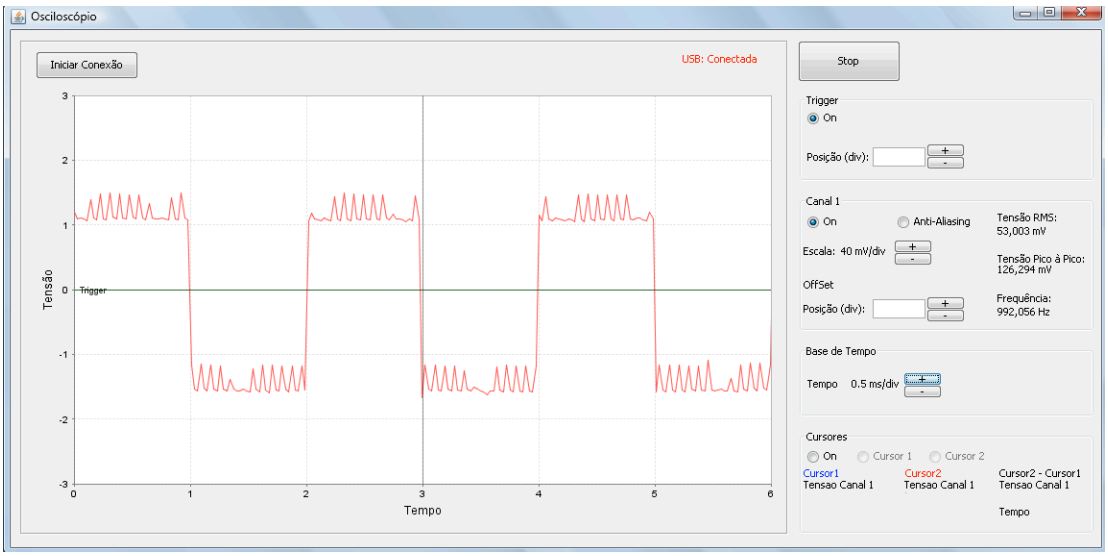


Figura 69 - Osciloscópio projetado [Vpp = 126,294 mV; Vrms = 53,003 mV; f = 992,056 Hz]

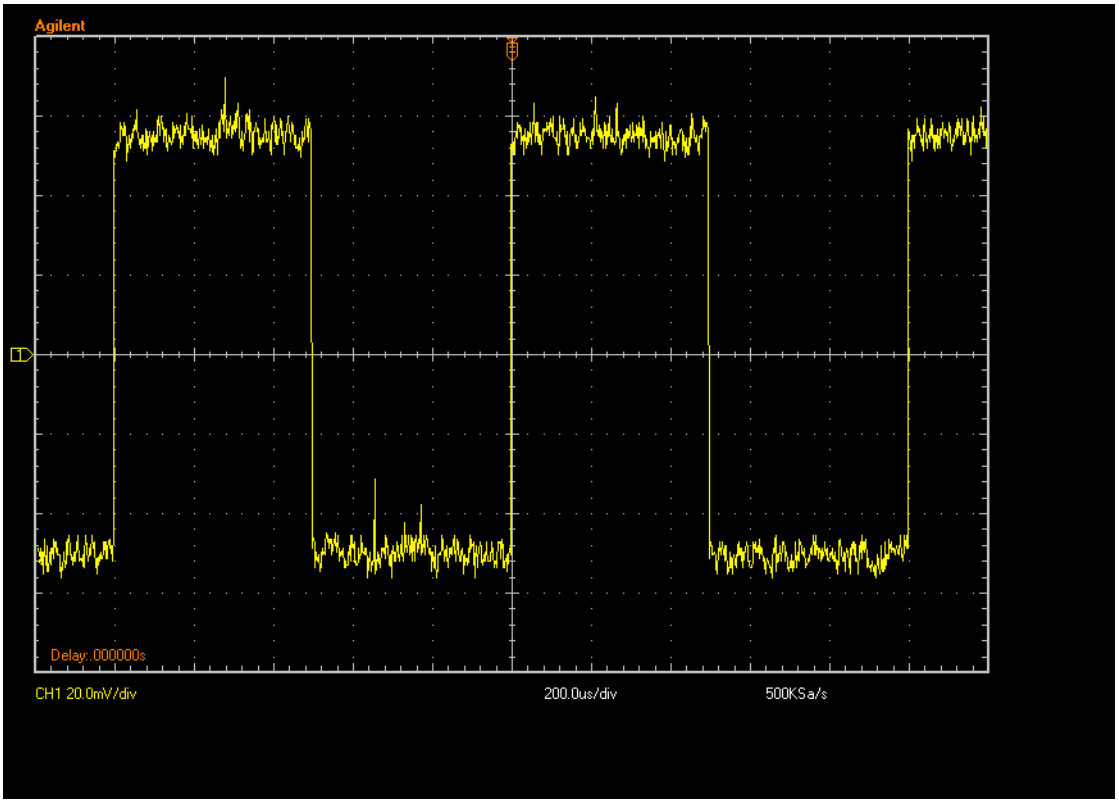


Figura 70 - Osciloscópio comercial [Vpp = 123 mV; Vrms = 52,8 mV; f = 1 kHz]

- Frequência 100 kHz e tensão 1,07 V

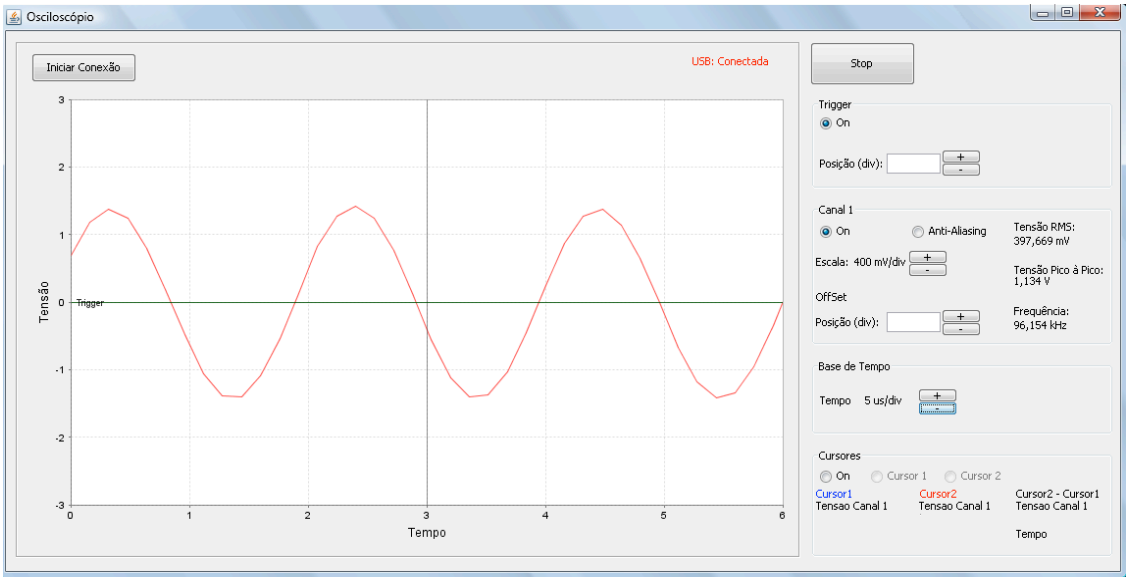


Figura 71 - Osciloscópio projetado [Vpp = 1,134 V; Vrms = 397,669 mV; f = 96,154 kHz]

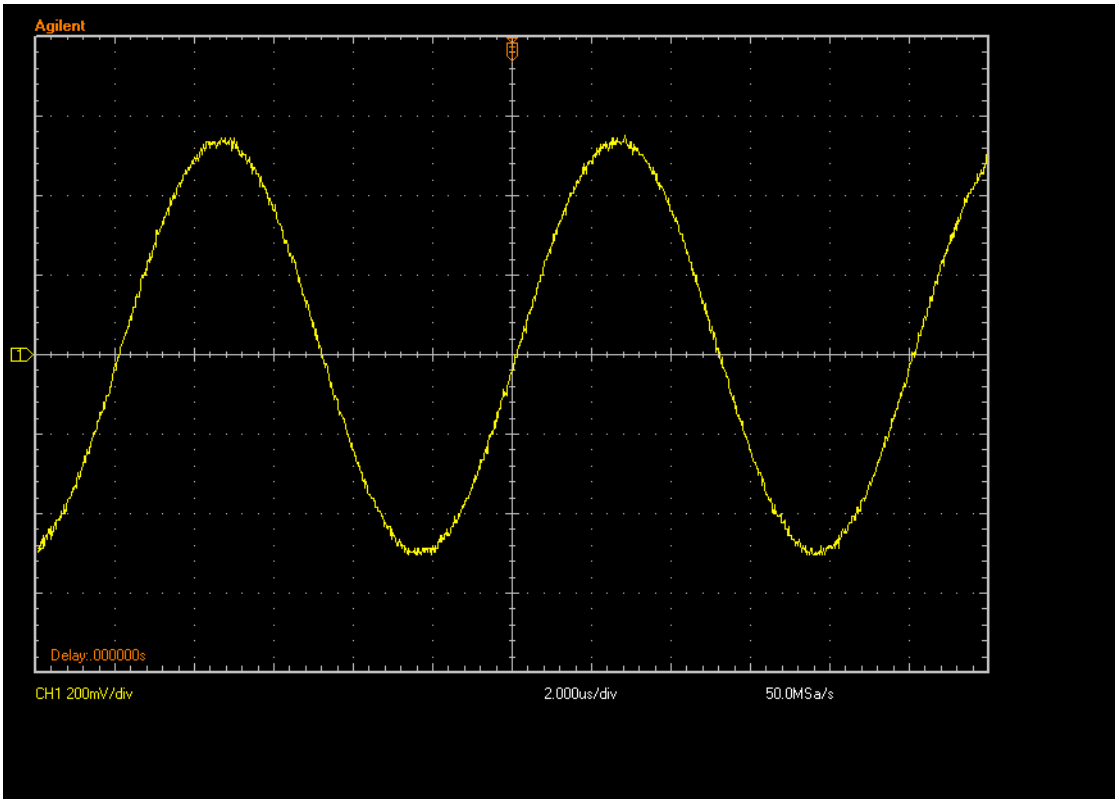


Figura 72 - Osciloscópio comercial [Vpp = 1,07 V; Vrms = 359 mV; f = 101 kHz]

- Frequência 1 Hz e tensão 2,08 V

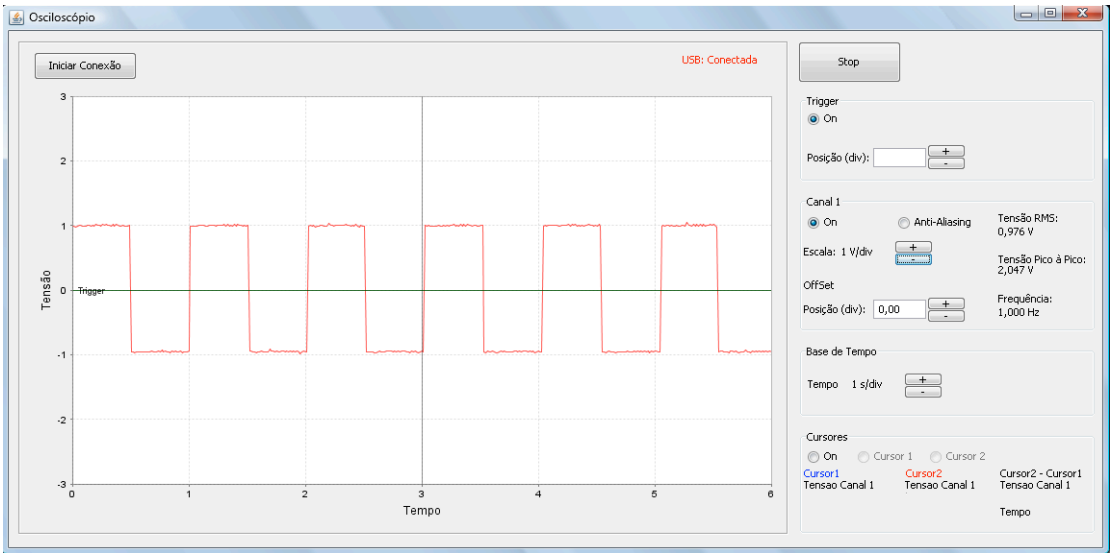


Figura 73 - Osciloscópio projetado [Vpp = 2,047 V; Vrms = 0,976 V; f = 1 Hz]

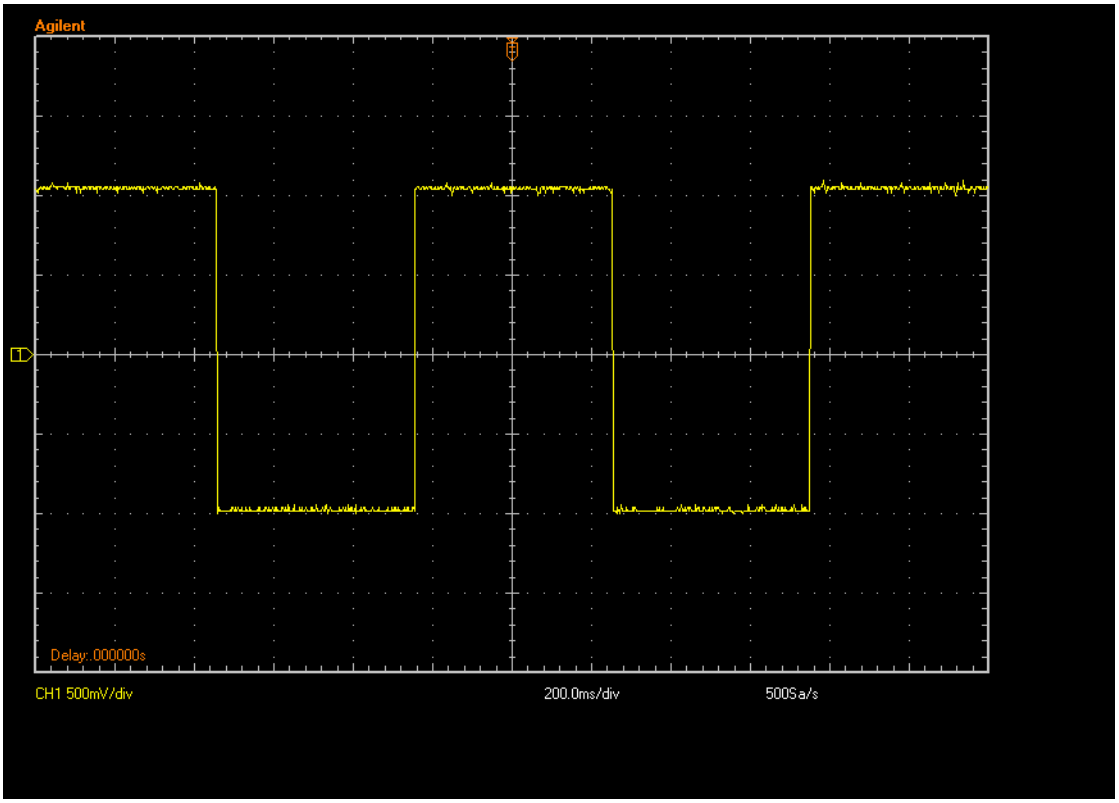


Figura 74 - Osciloscópio comercial [Vpp = 2,08 V; Vrms = 1,02 V; f = 1 Hz]

- Frequência 10 kHz com Tensão Pico a Pico 2,08 V e Ant-Aliasing ligado

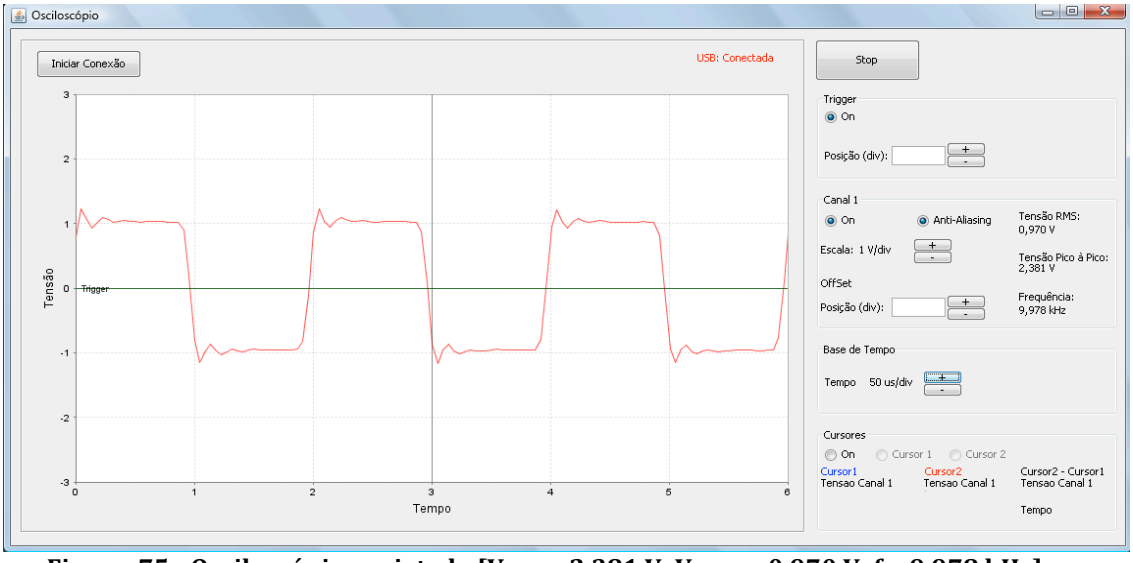


Figura 75 - Osciloscópio projetado [Vpp = 2,381 V; Vrms = 0,970 V; f = 9,978 kHz]

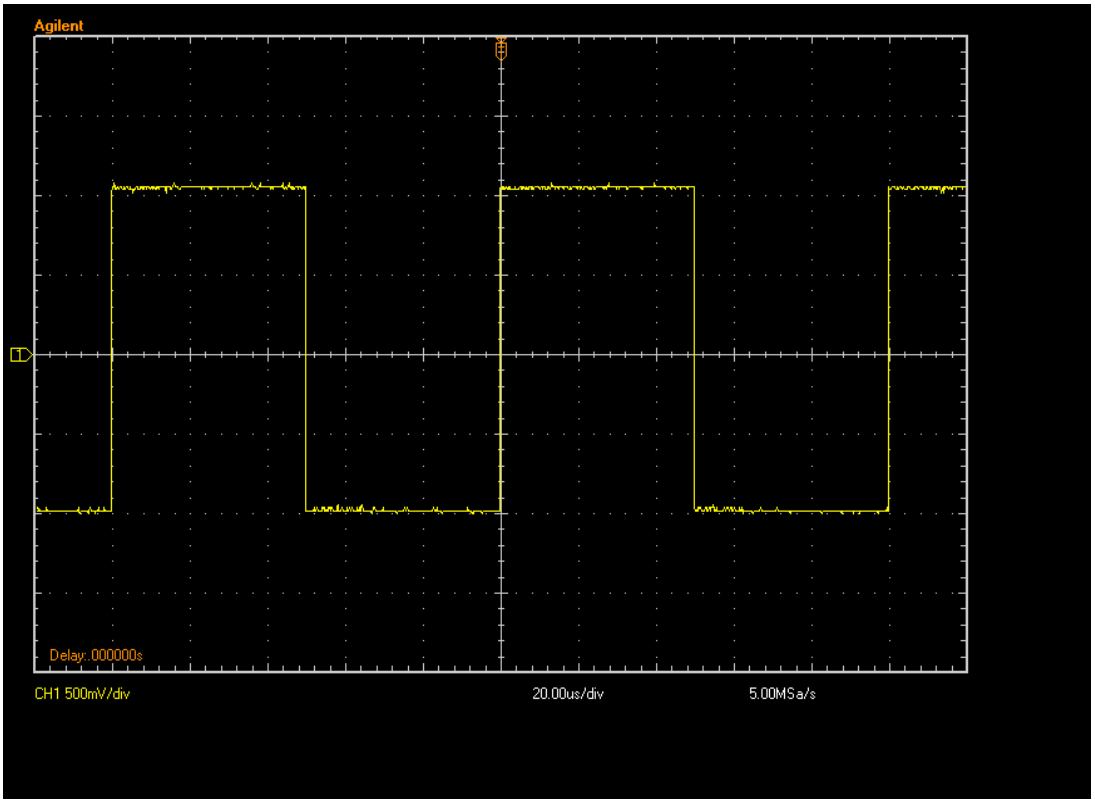


Figura 76 - Osciloscópio comercial [Vpp = 2,08 V; Vrms = 1,02 V; f = 10 kHz]

- Frequência 20 kHz com Tensão Pico a Pico 2,08 V e Ant-Aliasing ligado

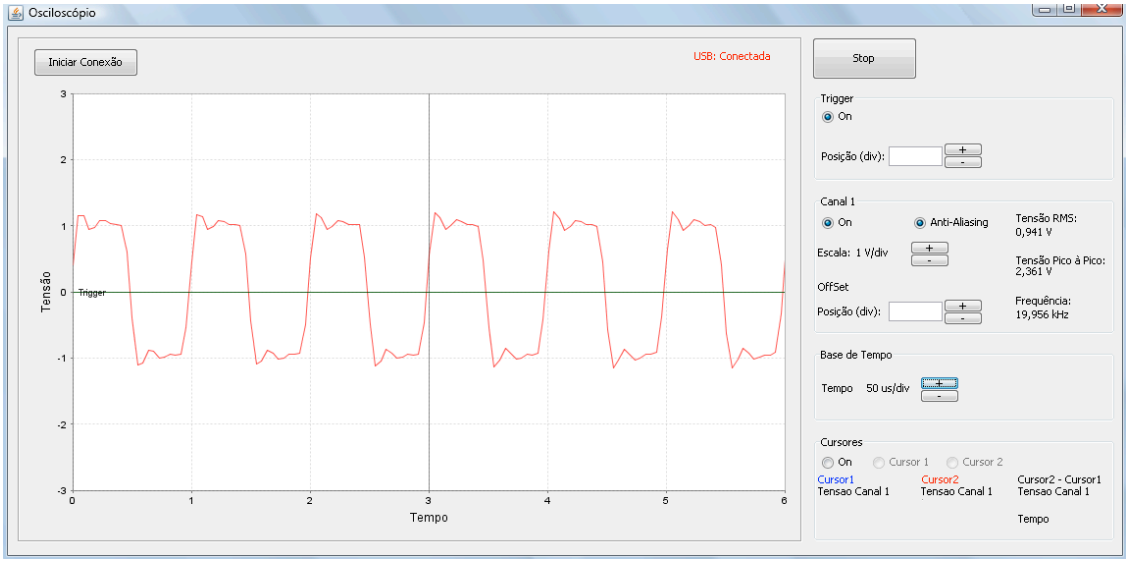


Figura 77 - Osciloscópio projetado [Vpp = 2,361 V; Vrms = 0,941 V; f = 19,956 kHz]

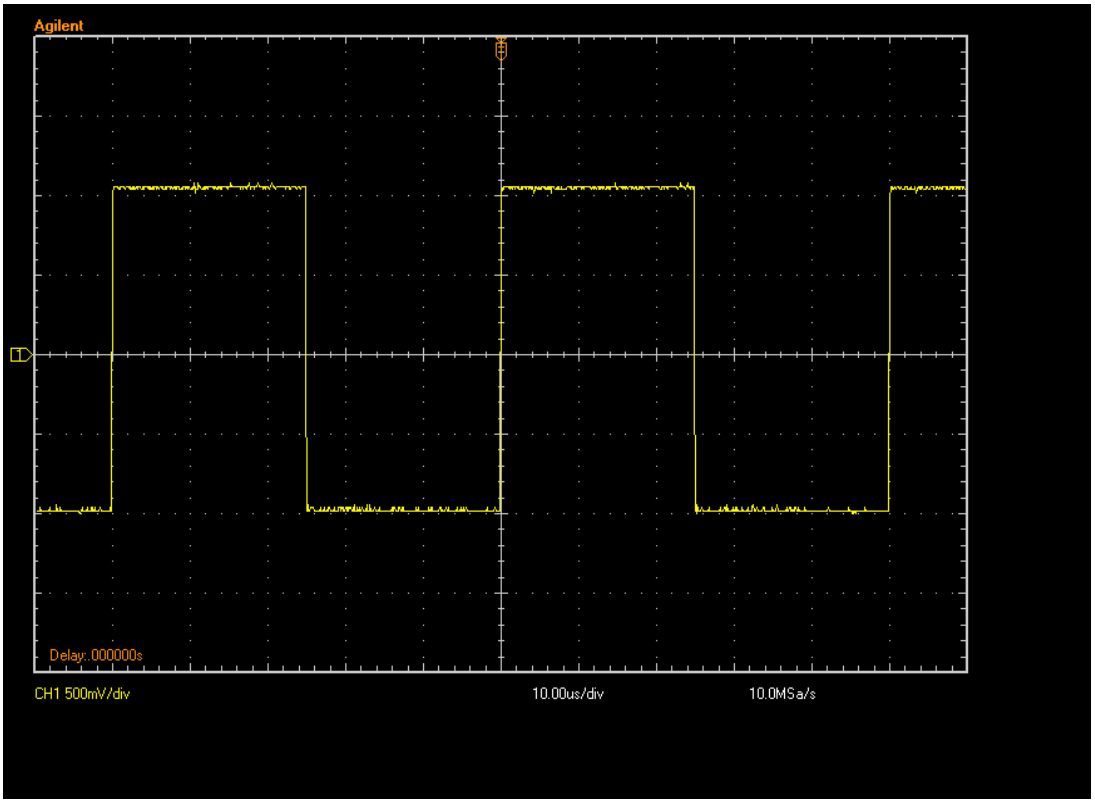


Figura 78 - Osciloscópio comercial [Vpp = 2,08 V; Vrms = 1,01 V; f = 20 kHz]

- Frequência 100 kHz com Tensão Pico a Pico 2,12 V e Ant-Aliasing ligado

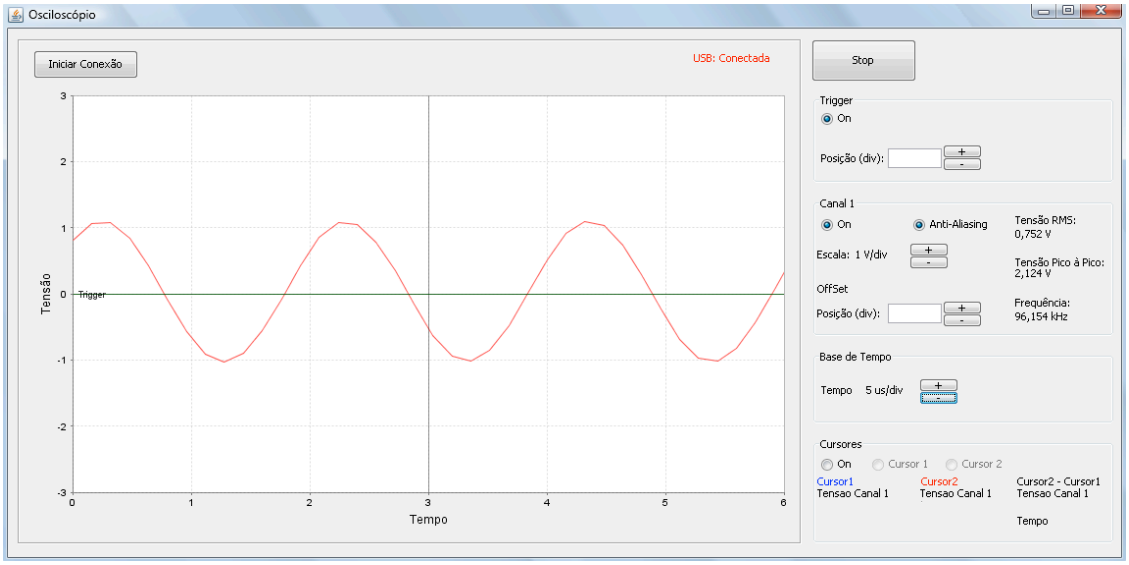


Figura 79 - Osciloscópio projetado [Vpp = 2,124 V; Vrms = 0,752 V; f = 96,154 kHz]

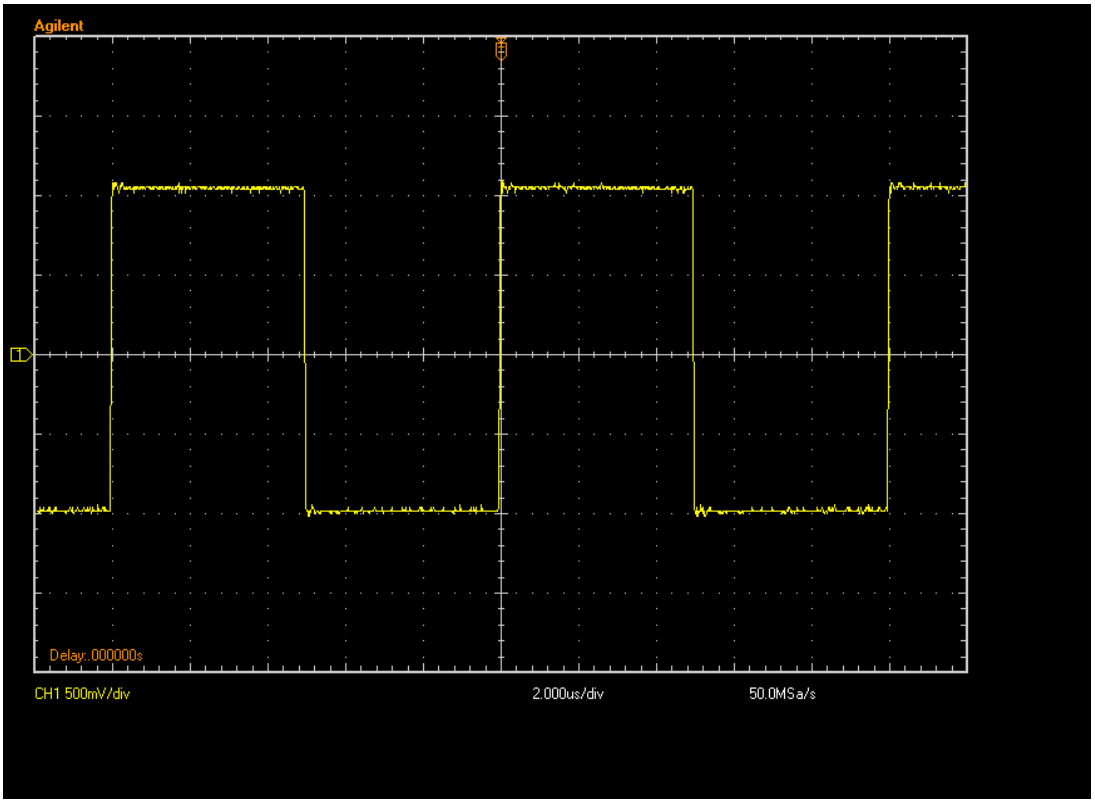


Figura 80 - Osciloscópio comercial [Vpp = 2,12 V; Vrms = 1,01 V; f = 100 kHz]

- Frequência 150 kHz com Tensão Pico a Pico 2,12 V e Ant-Aliasing ligado

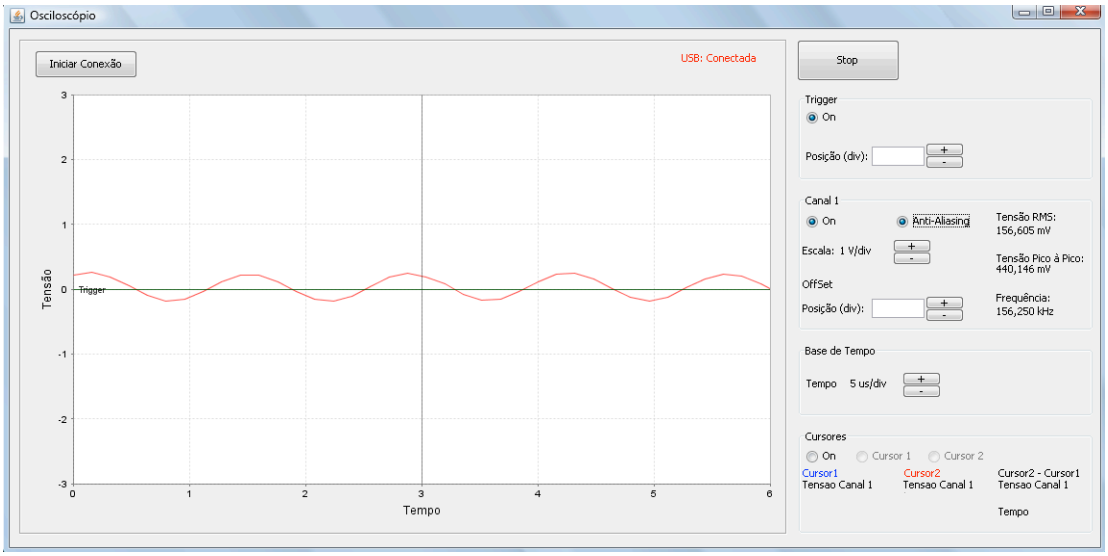


Figura 81 - Osciloscópio projetado [Vpp = 440,146 mV; Vrms = 156,605 mV; f = 156,250 kHz]

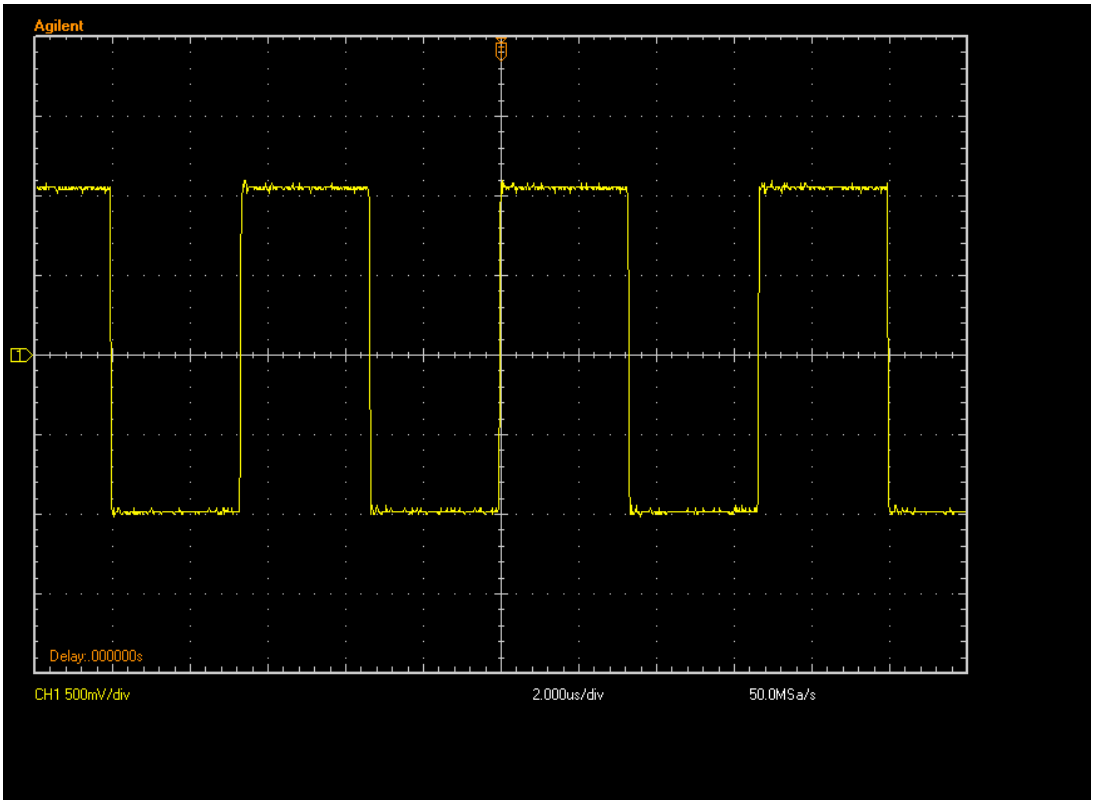


Figura 82 - Osciloscópio comercial [Vpp = 2,12 V; Vrms = 1,01 V; f = 150 kHz]

- Frequência 200 kHz com Tensão Pico a Pico 2,12 V e Ant-Aliasing ligado

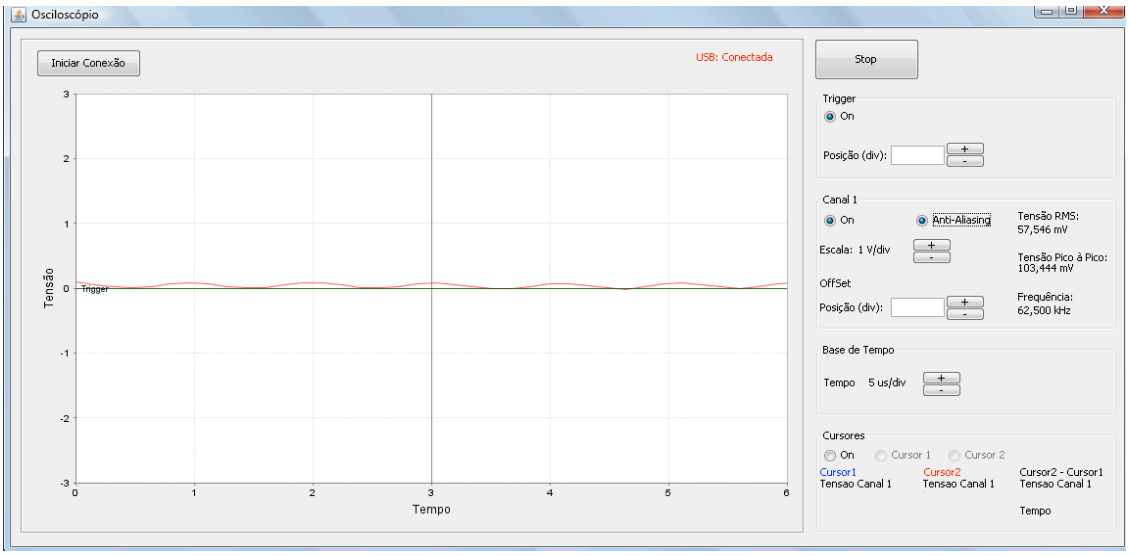


Figura 83 - Osciloscópio projetado [Vpp = 103,44 mV; Vrms = 57,546 mV; f = 62,500 kHz]

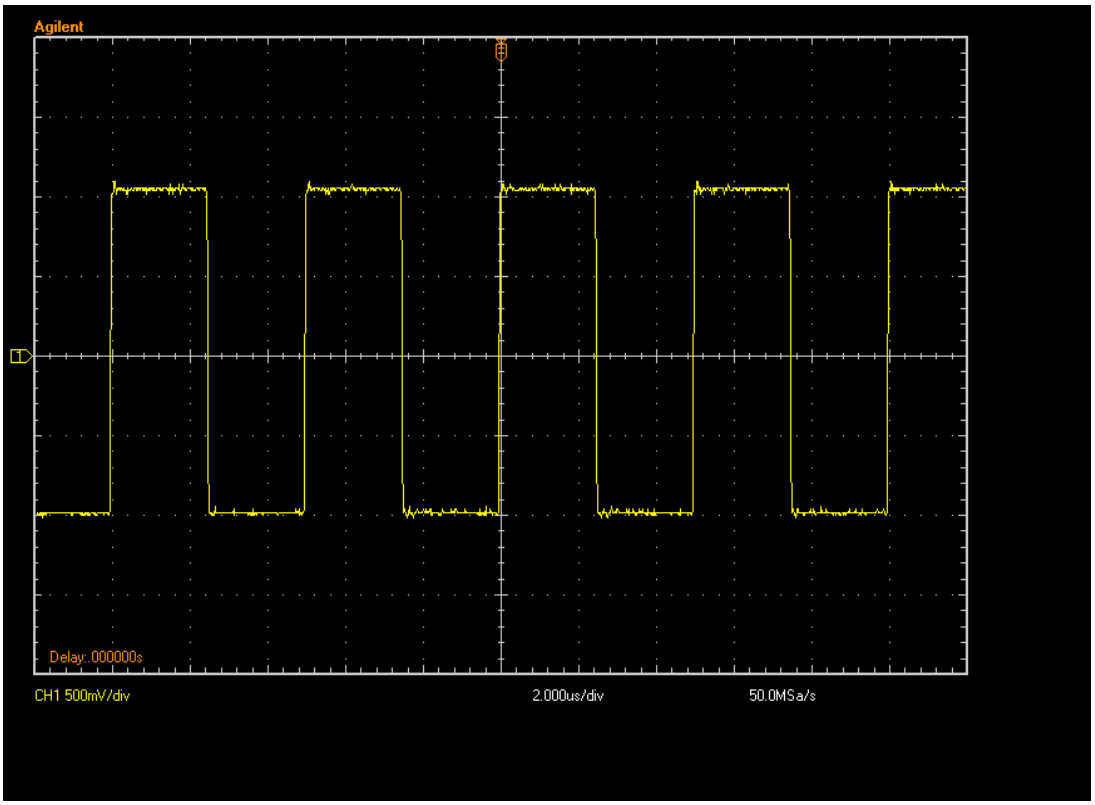


Figura 84 - Osciloscópio comercial [Vpp = 2,12 V; Vrms = 1,01 V; f = 200 kHz]

- Frequência 1Hz e Tensão Pico a Pico 20,2 V (Onda senoidal)
  A tensão pico a pico teve um erro de 2,2 %, a tensão rms teve um erro de 4,1% e a frequência um erro de 1,4%.
- Frequência 1 kHz e Tensão Pico a Pico 118 mV (Onda senoidal)
  A tensão pico a pico teve um erro de 3,2 %, a tensão rms teve um erro de 4,9% e a frequência um erro de 0,8%.
- Frequência 1 kHz e Tensão Pico a Pico 123 mV (Onda Quadrada)
  A tensão pico a pico teve um erro de 2,6 %, a tensão rms teve um erro de 4,1% e a frequência um erro de 0,8%.
- Frequência 100 kHz e tensão 1,07 V (Onda Senoide)
  A tensão pico a pico teve um erro de 5,9 %, a tensão rms teve um erro de 10,7% e a frequência um erro de 0,5%.
- Frequência 1 Hz e tensão 2,08 V (Onda Quadrada)
  A tensão pico a pico teve um erro de 1,6 %, a tensão rms teve um erro de 4,5% e a frequência foi exatamente igual.

O erro de tensão pico a pico considerando todas as medidas foi de 3,1% o erro de tensão RMS levando em consideração todas as medidas foi de 5,6% e o erro da frequência de todas as medidas feitas foi de 0,6%.

6.4.2 Resposta em frequência do anti-alising

A reposta em frequência do anti-alising é levantada a seguir:

- Frequência 10 kHz com Tensão Pico a Pico 2,08 V e Anti-Aliasing ligado (Onda Quadrada)
  O ganho foi de 1,168 dB.
- Frequência 20 kHz com Tensão Pico a Pico 2,08 V e Anti-Aliasing ligado (Onda Quadrada)
  O ganho foi de 1,099 dB.
- Frequência 100 kHz com Tensão Pico a Pico 2,12 V e Anti-Aliasing ligado (Onda Quadrada)
  O ganho foi de 0,008 dB.
- Frequência 150 kHz com Tensão Pico a Pico 2,12 V e Anti-Aliasing ligado (Onda Quadrada)
  O ganho foi de -13,61 dB.
- Frequência 200 kHz com Tensão Pico a Pico 2,12 V e Anti-Aliasing ligado (Onda Quadrada)
  O ganho foi de -26,375 dB.

Abaixo segue o diagrama de bode da resposta em frequência do Anti-Aliasing.

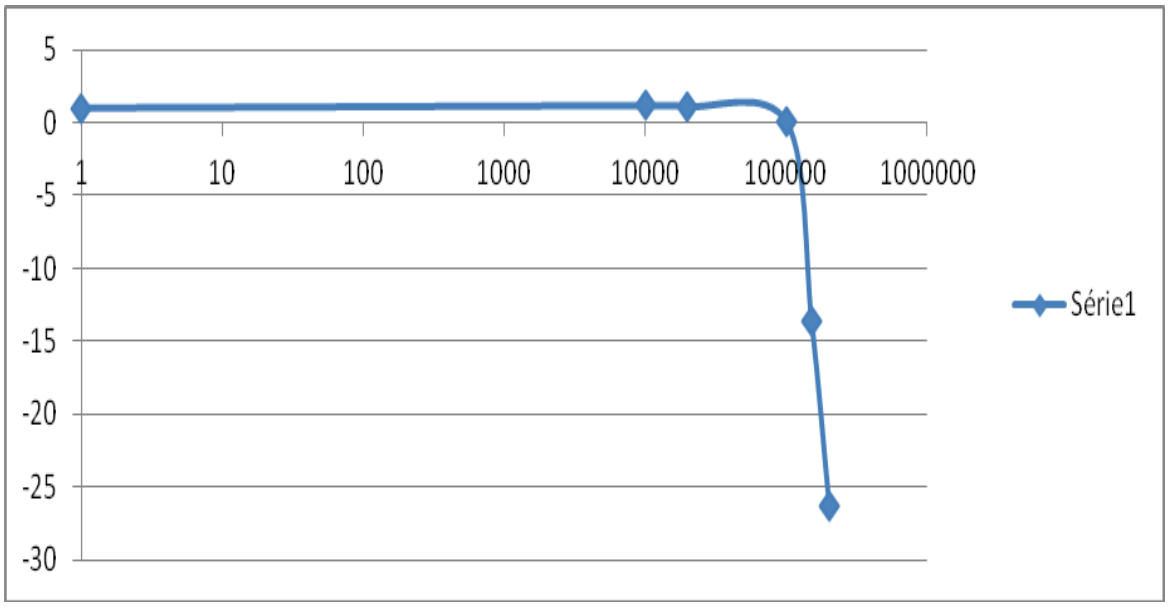


Figura 85 - Resposta em frequência do anti-alising

7 Cronograma

Apresentamos baixo o cronograma e o diagrama de Gantt atualizados:

	Nome	Duração	Início	Término	Prede...	Nome do Recurso	% Completa
1	Marco de Início	0 dias	14/09/11 08:00	14/09/11 08...			0%
2	**Execução**	60,5 dias?	**14/09/11 08:00**	**07/12/11 1...**			100%
3	**Estudo e Relátorio Técr**	6 dias	**14/09/11 08:00**	**21/09/11 1...**			100%
4	Sistema de Aquisição d	6 dias	14/09/11 08:00	21/09/11 17...		Tiago;Vinicius Zar...	100%
5	Microcontrolador	6 dias	14/09/11 08:00	21/09/11 17...		Geovane;Henrique	100%
6	Comunicação USB	6 dias	14/09/11 08:00	21/09/11 17...		Geovane;Tiago	100%
7	Modelagem de PCI	6 dias	14/09/11 08:00	21/09/11 17...		Tiago;Vinicius Zar...	100%
8	Funcionamento Ociloscó	6 dias	14/09/11 08:00	21/09/11 17...		Geovane;Henrique...	100%
9	Compra de Componente:	2 dias	14/09/11 08:00	15/09/11 17...		Vinicius Zaramella	100%
10	**Desenvolvimento**	54,5 dias?	**22/09/11 08:00**	**07/12/11 1...**	**3**		100%
11	**Hardware**	50,75 dias?	**22/09/11 08:00**	**01/12/11 1...**			100%
12	**PCI**	25,375 dias?	**27/10/11 08:00**	**01/12/11 1...**			100%
13	ProjetoMicrocontrola	20,25 dias?	31/10/11 10:00	28/11/11 13...			100%
14	pedidoMicrocontrola	0 dias	28/11/11 13:00	28/11/11 13...	13		0%
15	ProjetoSAD	8 dias	27/10/11 08:00	07/11/11 17...	21	Tiago;Vinicius Zar...	100%
16	PedidoSAD	1 dia	08/11/11 08:00	08/11/11 17...	15	Tiago;Vinicius Zar...	100%
17	MontagemSAD	2,75 dias	24/11/11 08:00	28/11/11 15...		Geovane;Tiago	100%
18	TesteSad	2 dias	28/11/11 15:00	30/11/11 15...	17	Geovane;Tiago	100%
19	Documentação	0,625 dias	30/11/11 15:00	01/12/11 11...	18	Geovane;Tiago	100%
20	**Microcontrolador**	4 dias	**22/09/11 08:00**	**27/09/11 1...**			100%
21	Projeto do Circuito	3 dias	22/09/11 08:00	26/09/11 17...		Geovane;Henrique	100%
22	Documentação	1 dia	27/09/11 08:00	27/09/11 17...	21	Geovane;Henrique	100%
23	**Sistema de Aquisiçã**	50,75 dias	**22/09/11 08:00**	**01/12/11 1...**			100%
24	Projeto	16 dias	22/09/11 08:00	13/10/11 17...		Tiago;Vinicius Zar...	100%
25	Teste	2 dias	28/11/11 15:00	30/11/11 15...	24;17	Tiago;Vinicius Zar...	100%
26	Documentação	1 dia	30/11/11 15:00	01/12/11 15...	25	Tiago;Vinicius Zar...	100%
27	**Comunicação**	27,125 dias	**24/10/11 08:00**	**30/11/11 0...**			100%
28	Estudo de Protocolos	3,5 dias	24/10/11 08:00	27/10/11 13...		Tiago;Vinicius Zar...	100%
29	Projeto do Protocolo	8,5 dias	27/10/11 12:00	08/11/11 17...	28	Tiago;Vinicius Zar...	100%
30	Implementação	9 dias	14/11/11 08:00	24/11/11 17...	29	Tiago;Vinicius Zar...	100%
31	Teste	2,125 dias	25/11/11 08:00	29/11/11 09...	30	Tiago;Vinicius Zar...	100%
32	Documentação	1 dia	29/11/11 09:00	30/11/11 09...	31	Tiago;Vinicius Zar...	100%
33	**Software**	48,75 dias?	**29/09/11 08:00**	**06/12/11 1...**			100%
34	**Estação Base**	43,5 dias?	**29/09/11 08:00**	**29/11/11 1...**			100%
35	Projeto	7,5 dias?	29/09/11 08:00	10/10/11 13...		Geovane;Henrique	100%
36	Implementação	27,5 dias	18/10/11 08:00	24/11/11 13...	35	Geovane;Henrique	100%
37	Teste	2 dias	24/11/11 12:00	28/11/11 13...	36	Geovane;Henrique	100%
38	Documentação	1 dia	28/11/11 12:00	29/11/11 13...	37	Geovane;Henrique	100%
39	**Sistema Embarcado**	20,75 dias	**08/11/11 08:00**	**06/12/11 1...**			100%
40	Projeto	0 dias	08/11/11 08:00	11/11/11 13...		Vinicius Zaramella	0%
41	Implementação	7,5 dias	18/11/11 12:00	29/11/11 17...	40	Vinicius Zaramella	100%
42	Teste	2,75 dias	30/11/11 08:00	02/12/11 15...	41	Vinicius Zaramella	100%
43	Documentação	2 dias	02/12/11 15:00	06/12/11 15...	42	Vinicius Zaramella	100%
44	**Sistema Final**	11,875 dias	**21/11/11 14:00**	**07/12/11 1...**			100%
45	Testes do Sistema Fin	11,875 dias	21/11/11 14:00	07/12/11 13...		Geovane;Henrique...	100%
46	Documentação de Tes	11,875 dias	21/11/11 14:00	07/12/11 13...		Geovane;Henrique...	100%
47	Relátorio Final	11,875 dias	21/11/11 14:00	07/12/11 13...		Geovane;Henrique...	100%
48	Criar Apresentação	11,625 dias	21/11/11 14:00	07/12/11 10...		Geovane;Henrique...	100%
49	Marco de Fim	0 dias	09/12/11 08:00	09/12/11 08...	44		0%

Figura 86 - Cronograma atualizado

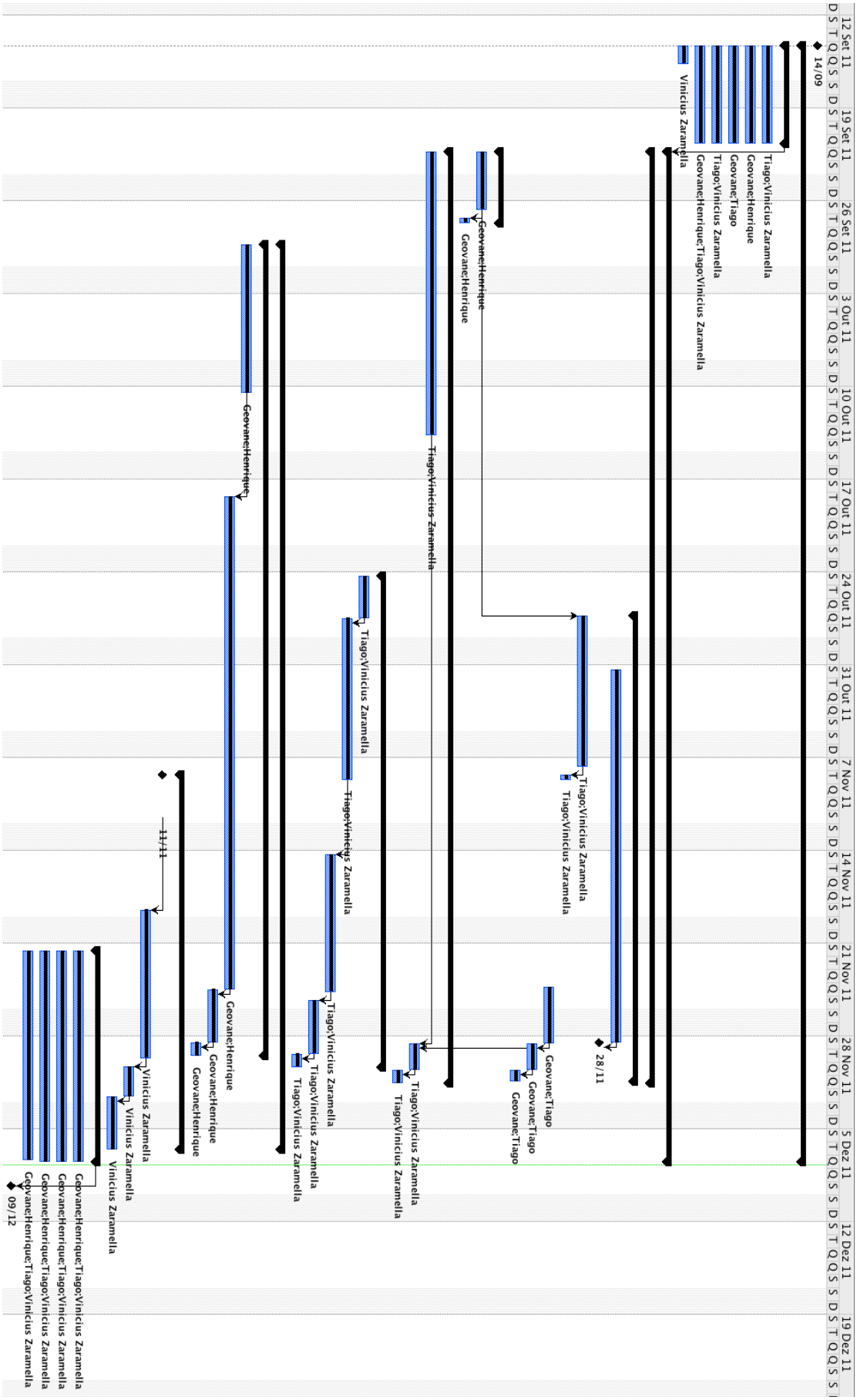


Figura 87 - Gantt atualizado

Custo do Projeto

Custo Do Projeto			
Componentes	Preço	Quantidade	Orçamento final
uC	R$ 32,00	2	R$ 64,00
kit de desenvolovimento	R$ 102,00	2	R$ 204,00
programador JTAG	R$ 41,75	1	R$ 41,75
PCI	R$ 180,00	1	R$ 180,00
Amplificador TL082	R$ 2,00	6	R$ 12,00
Fonte Chaveada	R$ 2,30	3	R$ 6,90
Resistores	10R$	100	R$ 10,00
Capacitores	10R$	100	R$ 10,00
Transistores	5R$	2	R$ 10,00
Diodos	2R$	2	R$ 4,00
Cristal	5R$	2	R$ 10,00
Regulador de Tensão (LD1117)	5R$	2	R$ 10,00
Chave Táctil	2R$	4	R$ 8,00
Conector USB	5R$	1	R$ 5,00
Relé SH1NAC2	1.5R$	6	R$ 9,00
LT1054	R$ 8,00	1	R$ 8,00
Frete	R$ 150,00	1	R$ 150,00
TOTAL:			R$ 742,65

Tabela 10 - Custo do gasto com o Projeto

Custo do protótipo utilizado no Projeto		
Componente	Preço (total)	Unidades
Placa de Circuito Impresso (Castaldo)	150+30(extra)R$	1
Kit de Desenvolvimento STM32F103	102R$	1
Resistores	5R$	45
Capacitores	10R$	72
Relé SH1NAC2	10R$	6
ULN2001D	2R$	1
Barra de Pinos	1R$	1
TL082	12R$	6
1N4001	5R$	5
LT1054	8R$	1
TOTAL:	187R$	

Tabela 11 - Custo referente ao protótipo que a equipe utilizou.

Custo do protótipo - Placa integrada		
Parte I - Condicionamento de Sinal		
Componente	Preço (total)	Unidades
Placa de Circuito Impresso (Castaldo)	200+30(extra)R$	1
Resistores	5R$	45
Capacitores	10R$	72
Relé SH1NAC2	10R$	6
ULN2001D	2R$	1
TL082	12R$	6
1N4001	5R$	5
LT1054	8R$	1
Parte 2 - Microcontrolador		
Microcontlador	32R$	1
Resistores	5R$	27
Capacitores	5R$	21
Transistores	5R$	2
Diodos	2R$	1
Cristal	5R$	1
Regulador de Tensão (LD1117)	5R$	1
Chave Táctil	2R$	2
Conector USB	5R$	1
TOTAL:	150R$	

Tabela 12 - Custo referente a um protótipo integrado (microcontrolador e condicionamento de sinal na mesma PCI).

8 Conclusões

8.1 ANÁLISE DO DESENVOLVIMENTO

Em relação ao planejamento de riscos, os riscos planejados e que de fato ocorreram foram os seguintes:

- Escolha de tecnologia inadequada para atender os requisitos do cliente.

Esse risco acabou ocorrendo em duas situações ao longo do projeto, em relação à comunicação do microcontrolador e estação base na qual teve que ser modificada de USB para serial, essa mudança foi explicada no tópico na seção 5.2.2, onde é detalhado também seu motivo.

A outra situação em que ocorreu esse risco foi na escolha do microcontrolador que não havia suporte DMA para os dois ADCs, porém nessa situação não houve mudança de tecnologia, pois já tínhamos adquirido o microcontrolador que não atendia todos os requisitos do projeto, e, além disso, tivemos conhecimento dessa incapacidade do microcontrolador somente próximo ao final do projeto.

- Subestimação da complexidade, ou dificuldade na implementação de software ou hardware

Ao longo do projeto percebemos que algumas etapas foram subestimadas,

a etapa que mais atrasou o progresso do projeto foi o desenvolvimento do circuito de condicionamento de sinal, que ficou realmente pronto após o segundo deliverable.

Outra etapa que atrasou o desenvolvimento do projeto foi a comunicação do sistema embarcado com estação-base, em que tivemos problemas com a tecnologia escolhida como descrito na seção 5.2.2

- Falha na comunicação entre estação base e o microcontrolador.

Situação explicada no primeiro risco analisado.

- Não cumprimento dos prazos das metas estabelecidas pela equipe.

A entrega de alguns deliverables foram atrasadas e ocorreram nas seguintes situações:

- Primeiro Deliverable

  O circuito de aquisição de dados não havia sido ainda definido, portanto não ocorreram testes, havia apenas algumas ideias de como projetá-lo.

- Segundo Deliverable

  O projeto da placa de circuito impresso nesse deliverable estava começando, o motivo desse atraso foi que o a subestimação no prazo que a equipe deu para o desenvolvimento do circuito de condicionamento de sinal

- Terceiro Deliverable

  O circuito em que testamos a aquisição de dados foi um montado em protoboard, a PCI havia sido pedida, mas ainda não tínhamos ela em mãos.

- Quarto Deliverable

  O protocolo de comunicação não havia sido ainda implementado em nenhum dos dois sistemas (estação base e sistema embarcado), além disso o software da estação-base e do microcontrolador estavam incompletos.

- Erro de planejamento

O erro de planejamento da equipe foi colocar semanas de folga somente no fim do projeto, e não entre as tarefas e, portanto, distribuindo essa folga, apesar disso o projeto pode ser concluído até o fim das 10 semanas de desenvolvimento.

8.2 INTEGRAÇÃO

Os conhecimentos integrados de outras disciplinas empregados no projeto foram:

- Eletrônica II

  Na disciplina de eletrônica em que aprendemos o uso de amplificadores operacionais e filtros ativos, além disso, o funcionamento de um ADC.

- Programação

  As disciplinas de programação deram a base do conhecimento para os alunos para a implementação do sistema na estação base

- Comunicação de Dados e Redes De Computadores

  Utilizando os conhecimentos de protocolos de comunicação aprendidos ao longo do semestre nessas duas matérias, a equipe implementou o protocolo do projeto.

- Microcontroladores

  A disciplina de microcontroladores ajudou os membros da equipe a entender o funcionamento do microcontrolador bem como seus periféricos.

- Arquitetura e Organização de Computadores

  A disciplina de arquitetura e organização de computadores ensinou conceitos como DMA e acesso à memória.

- Analise e Projeto de Sistemas e Engenharia de Software

  Com a disciplina as duas disciplinas mencionadas acima a equipe aprendeu a fazer documentação de um software utilizando diagramas de classes, de casos de uso, de máquinas de estado, e de sequência, além disso os requisitos funcionais e não funcionais de um software.

8.3 TRABALHOS FUTUROS

Requisitos não atendidos ao término do projeto:

- Sistema possuindo dois canais.
  O motivo de não ser implementado este requisito foi devido ao DMA do microcontrolador possuir suporte a apenas um dos ADCs.

- Software com a função de single-shot.
  Inicialmente foi pensado em utilizar o segundo canal para fazer o single-shot no plotador, como não houve a implementação do sistema com dois canais, não foi possível fazer a função de single-shot.

- Montagem do protótipo em uma PCI possuindo microcontrolador e sistema de condicionamento de sinal.
  Devido ao tempo escasso não foi possível atender este requisito, a equipe somente projetou e encomendou uma PCI com o condicionamento de sinal.

- Alimentação do circuito de condicionamento de sinal via USB
  O charge pump comprado pela equipe (TL082) na forma que foi implementado gerava um ruído de até 200mV no sistema, tornando-se inviável a utilização deste, o circuito de condicionamento de sinal foi então alimentado por uma fonte externa.

- Garantir a medição de sinais com até 10mV na entrada
  O motivo de não implementar este requisito é que a própria fonte de alimentação externa utilizada para realizar o projeto possuía um ruído de 20mV.

Requisitos que podem se incrementados à proposta inicial

- Uma das ideias discutidas no fim do projeto entre os membros era o acoplamento de um gerador de funções ao osciloscópio.

Esse requisito está relacionado à motivação do projeto, pois caso os alunos de graduação em Engenharia Eletrônica e Computação possuíssem um gerador de funções acoplado ao osciloscópio, poderiam realizar experimentos de laboratórios em casa.

9 Anexos

9.1 COMPARAÇÃO COM OUTROS OSCILOSCÓPIOS

- Frequência 10 kHz e tensão 20,4 V

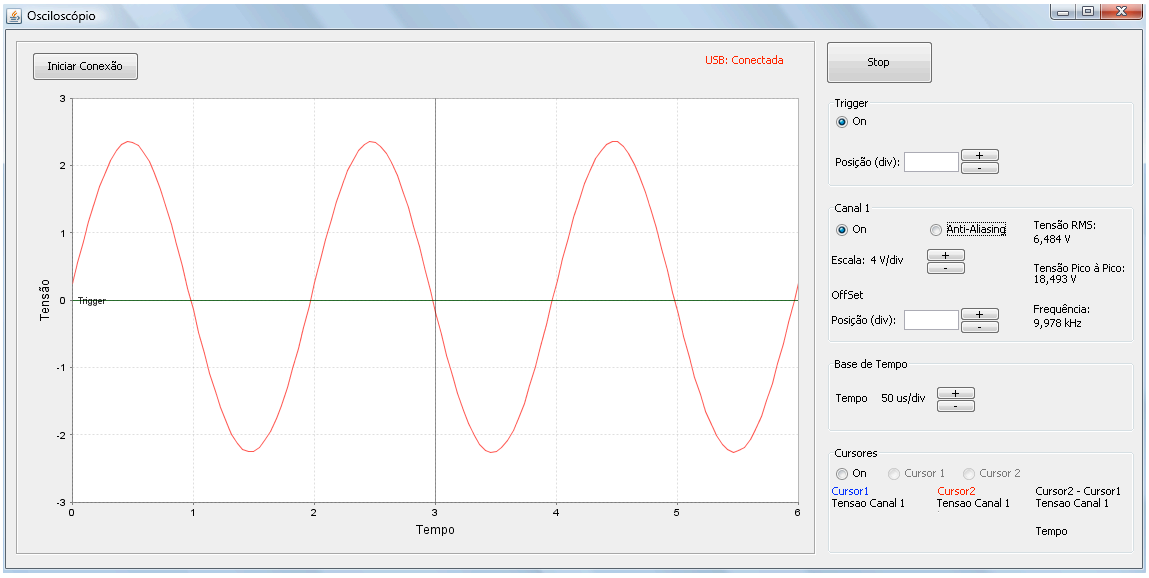


Figura 88 - Osciloscópio projetado [Vpp = 18,493 V; Vrms = 6,484 V; f = 9,978 kHz]

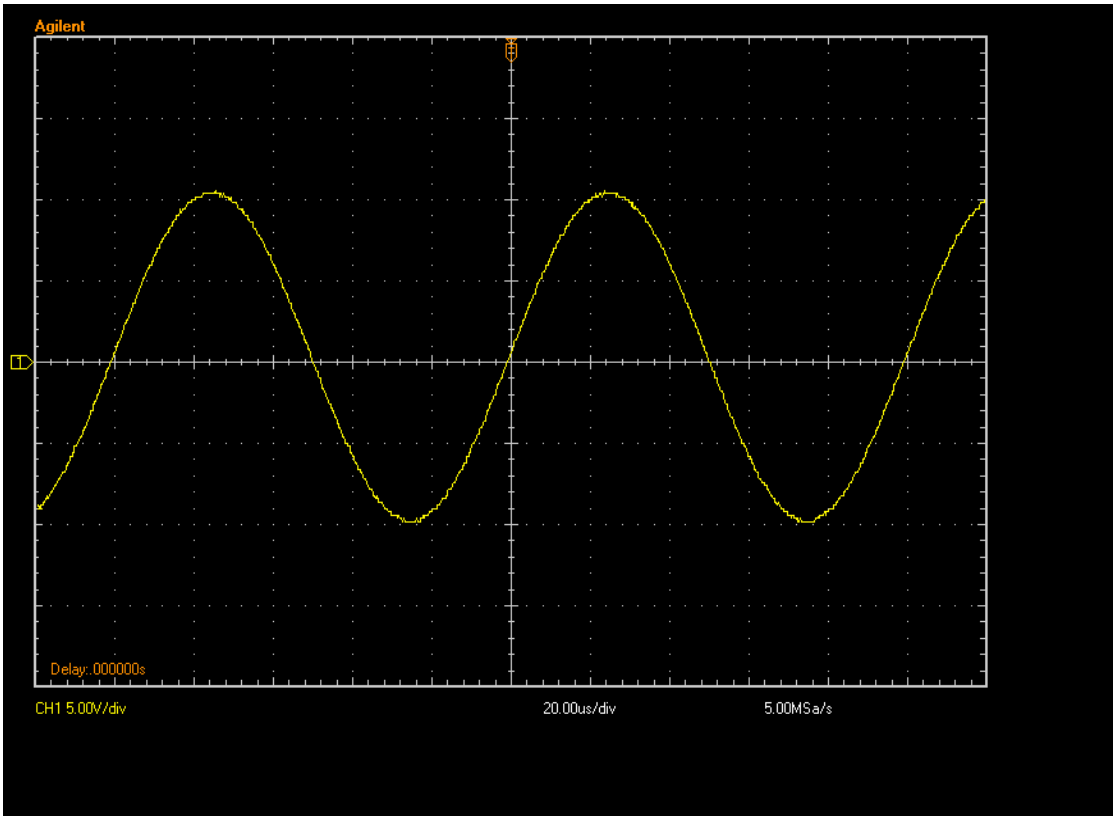


Figura 89 - Osciloscópio comercial [Vpp = 20,4 V; Vrms = 6,99 V; f = 10 kHz]

- Frequência 1 kHz e tensão 15,2 V

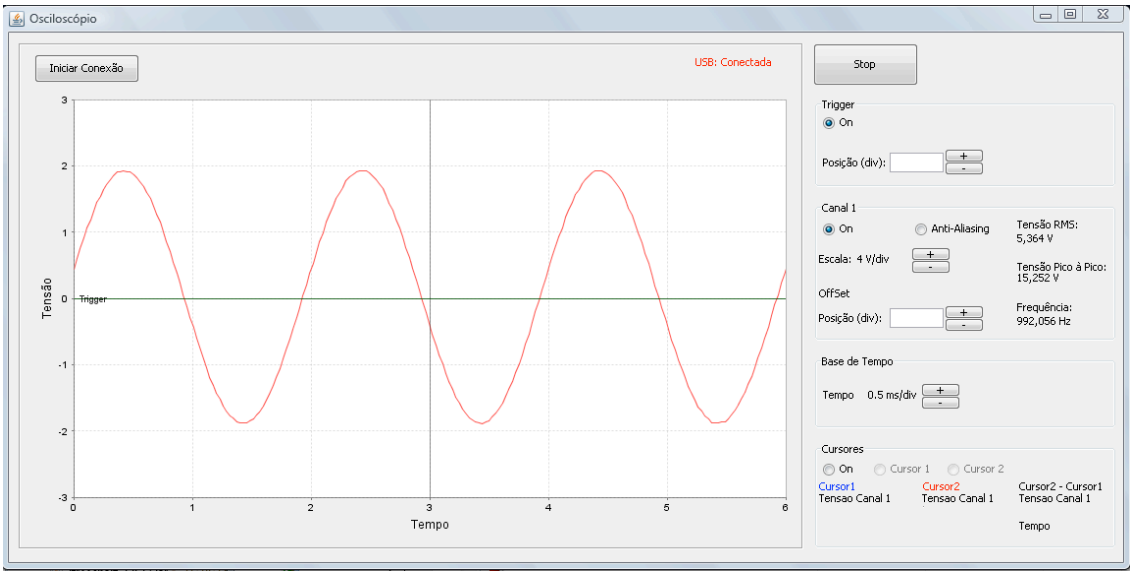


Figura 90 – Osciloscópio projetado [Vpp = 15,252 V; Vrms = 5,364 V; f = 992,056 Hz]

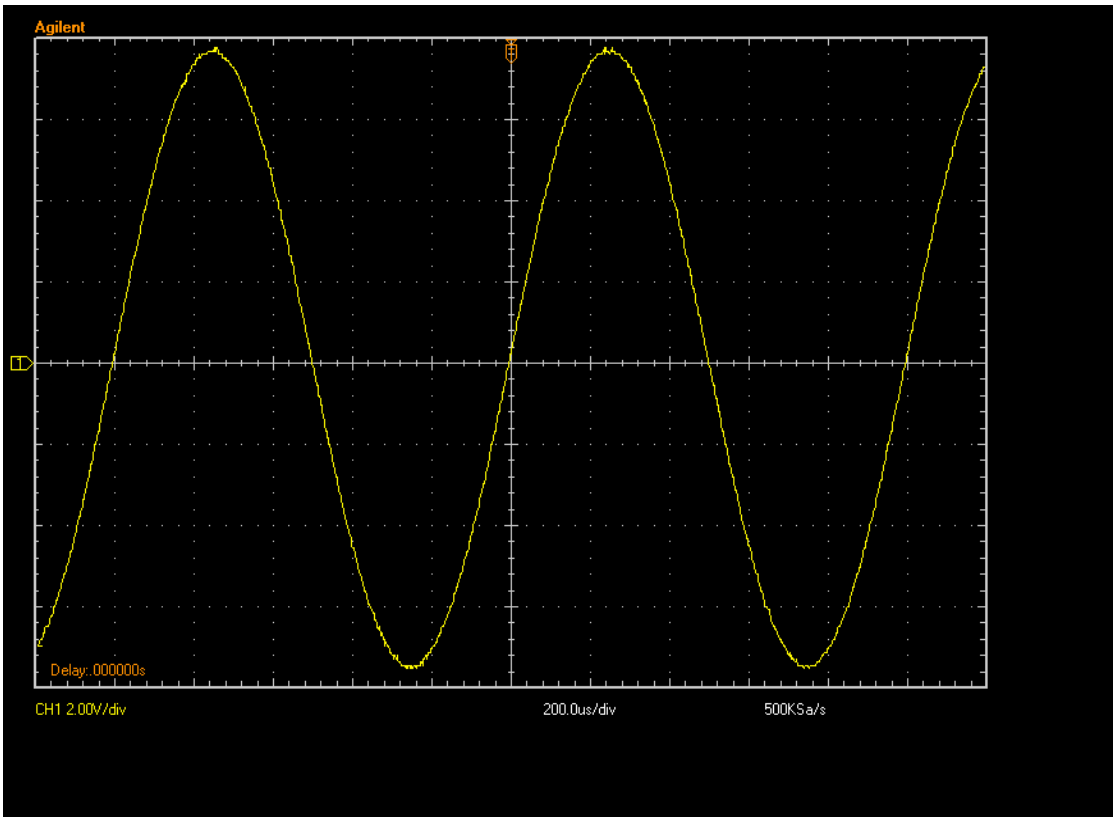


Figura 91 - Osciloscópio comercial [Vpp = 15,2 V; Vrms = 5,25 V; f = 1kHz]

- Frequência 1,01 Hz e tensão 15,4V

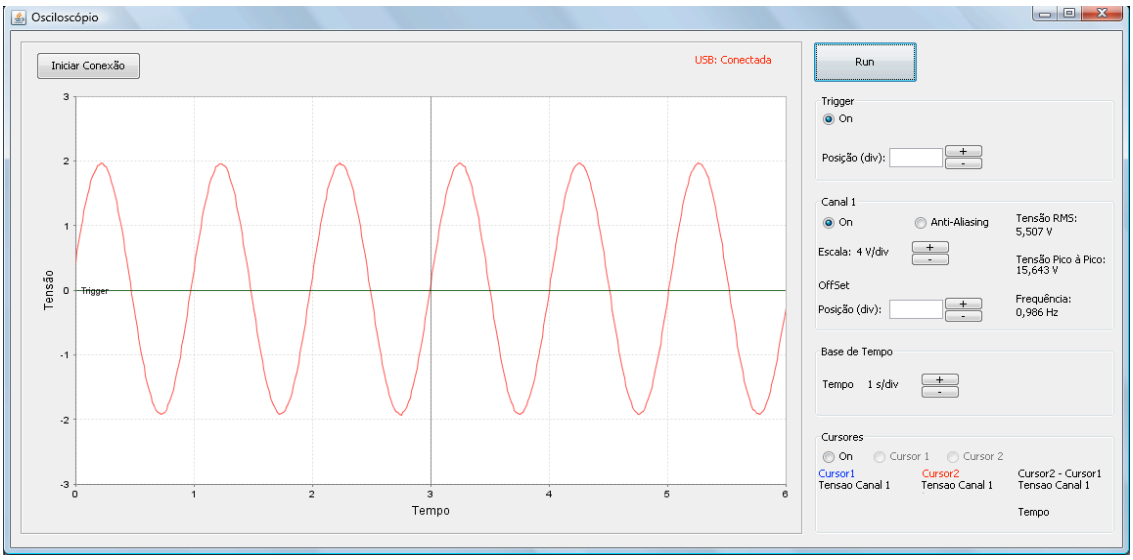


Figura 92 - Osciloscópio projetado [Vpp = 15,643 V; Vrms = 5,507 V; f = 0,986Hz]

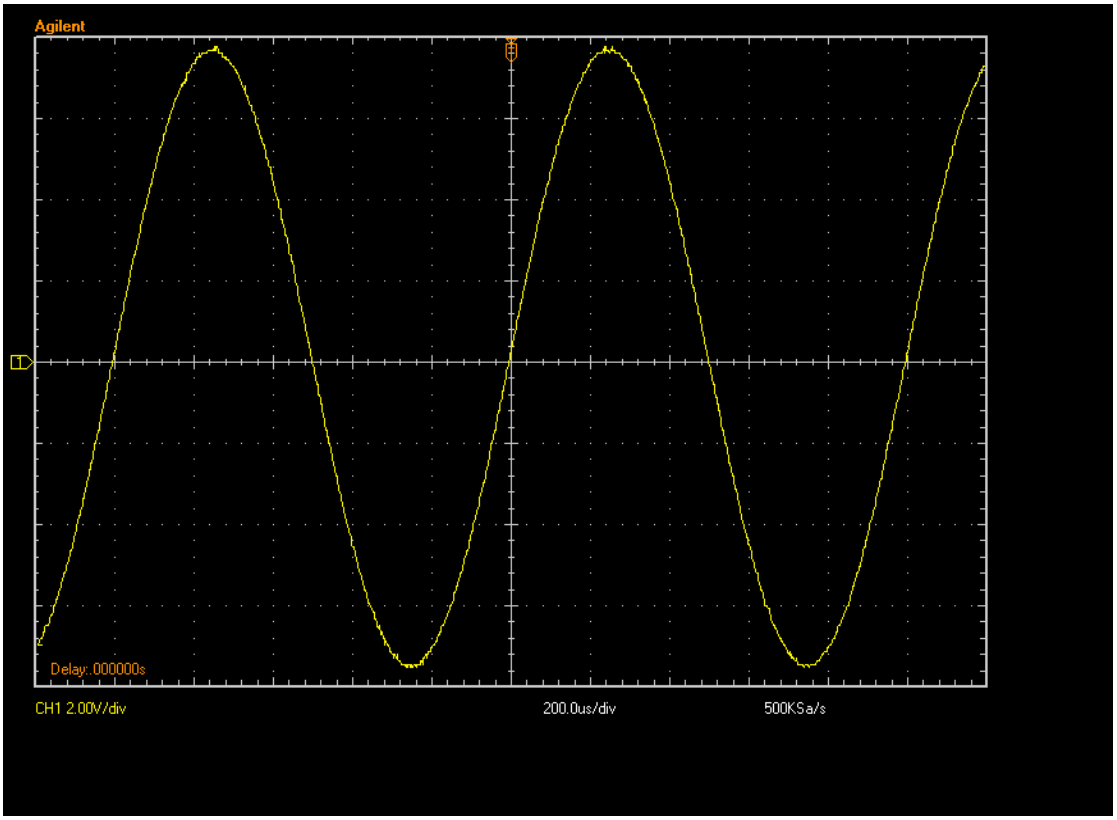


Figura 93 - Osciloscópio comercial [Vpp = 15,4 V; Vrms = 5,28 V; f = 1,01Hz]

- Frequência 10 kHz e tensão 15,2V

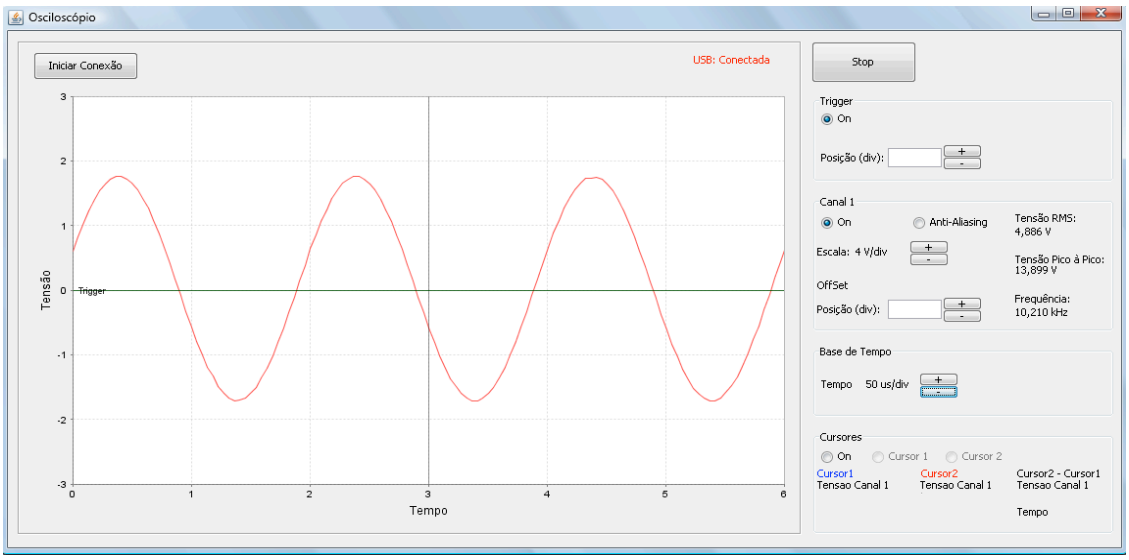


Figura 94 - Osciloscópio projetado [Vpp = 13,899 V; Vrms = 4,899 V; f = 10,210 kHz]

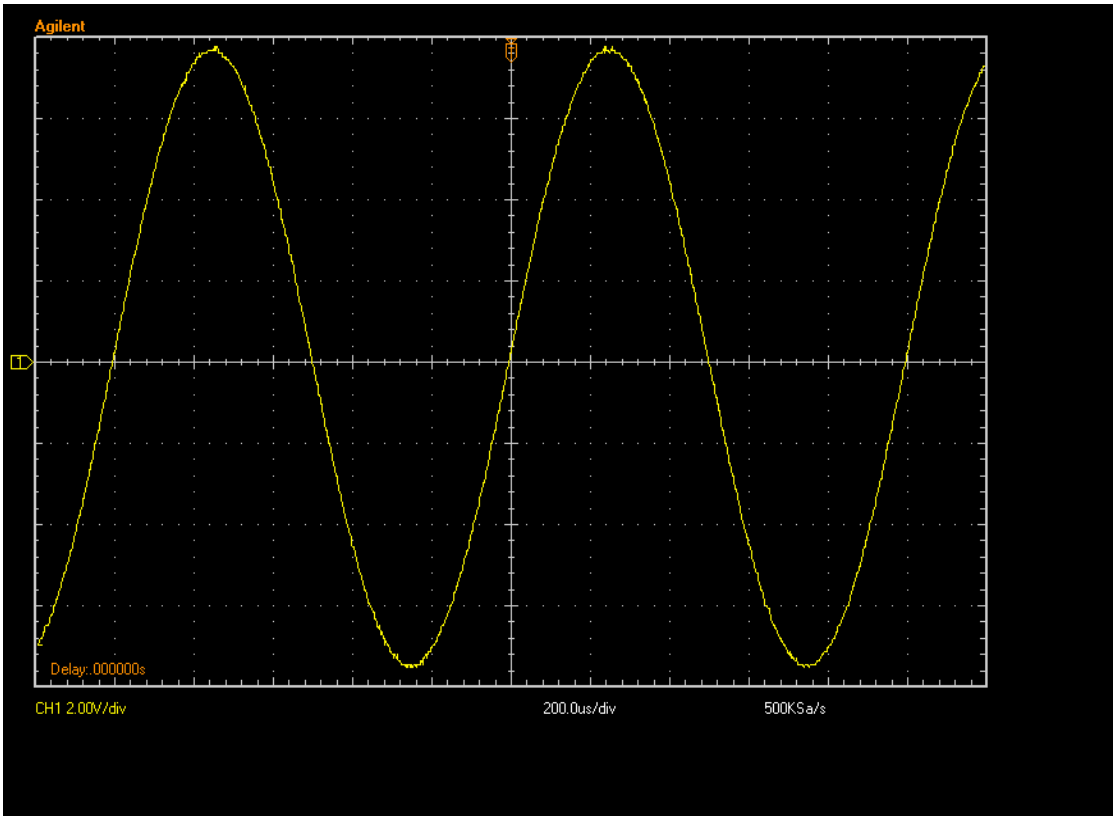


Figura 95 - Osciloscópio comercial [Vpp = 15,2 V; Vrms = 5,24 V; f = 10 kHz]

- Frequência 100 kHz e tensão 15,1 V

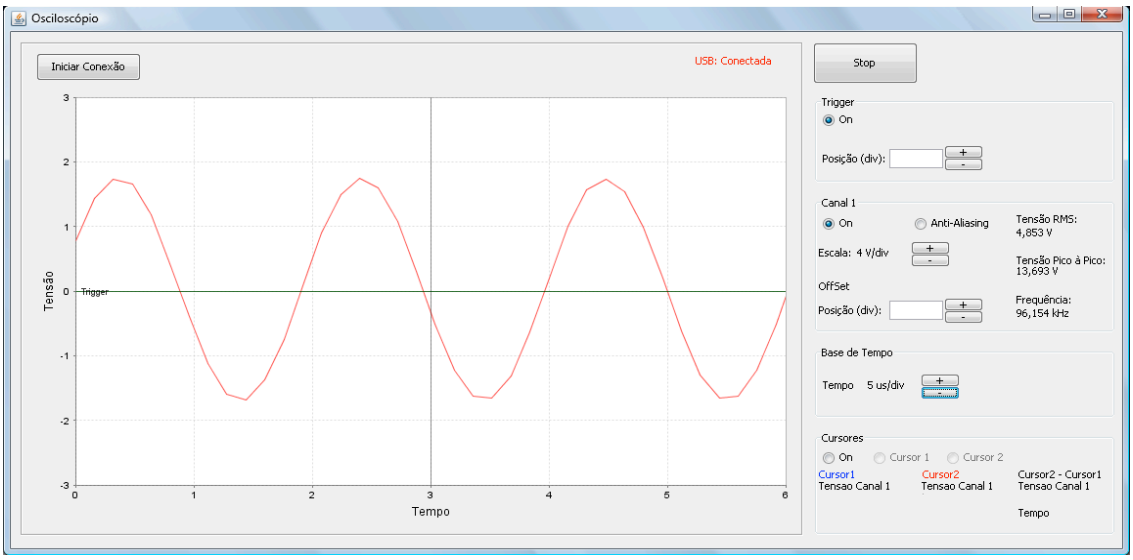


Figura 96 - Osciloscópio projetado [Vpp = 13,693 V; Vrms = 4,853 V; f = 96,154 kHz]

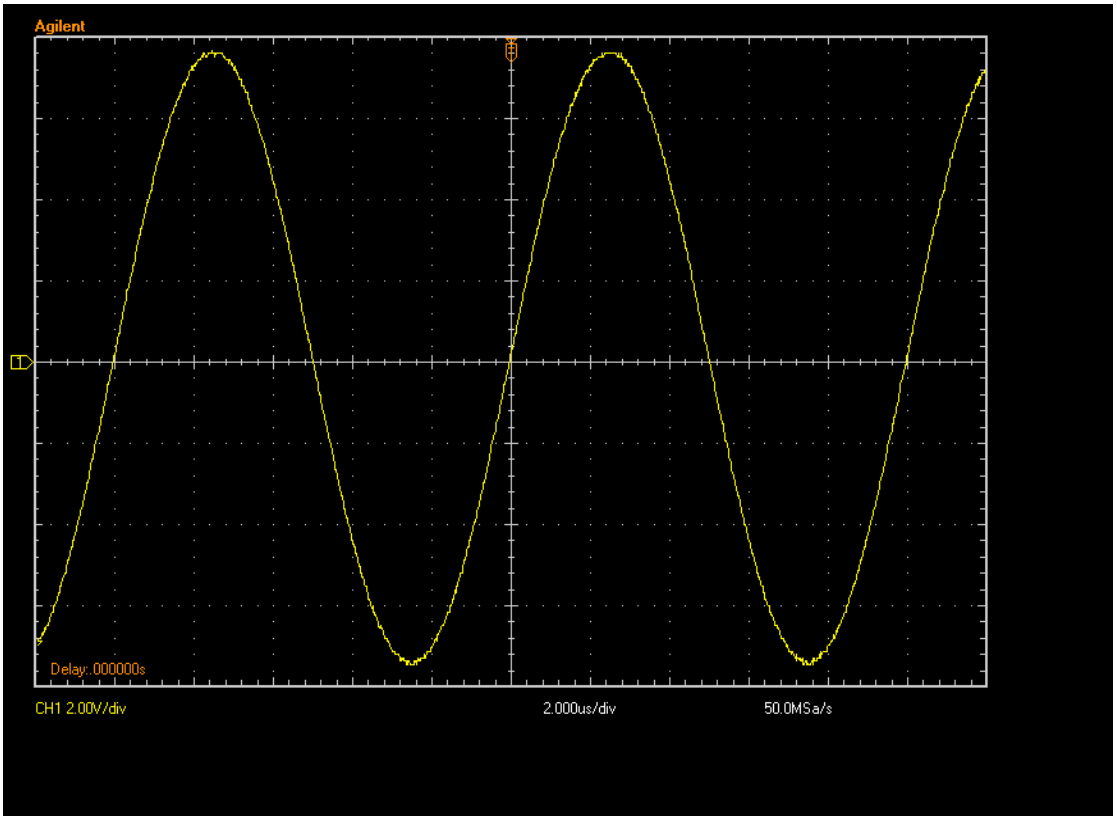


Figura 97 - Osciloscópio comercial [Vpp = 15,1 V; Vrms = 5,21V; f = 100 kHz]

- Frequência 100 kHz e tensão 15,7 V

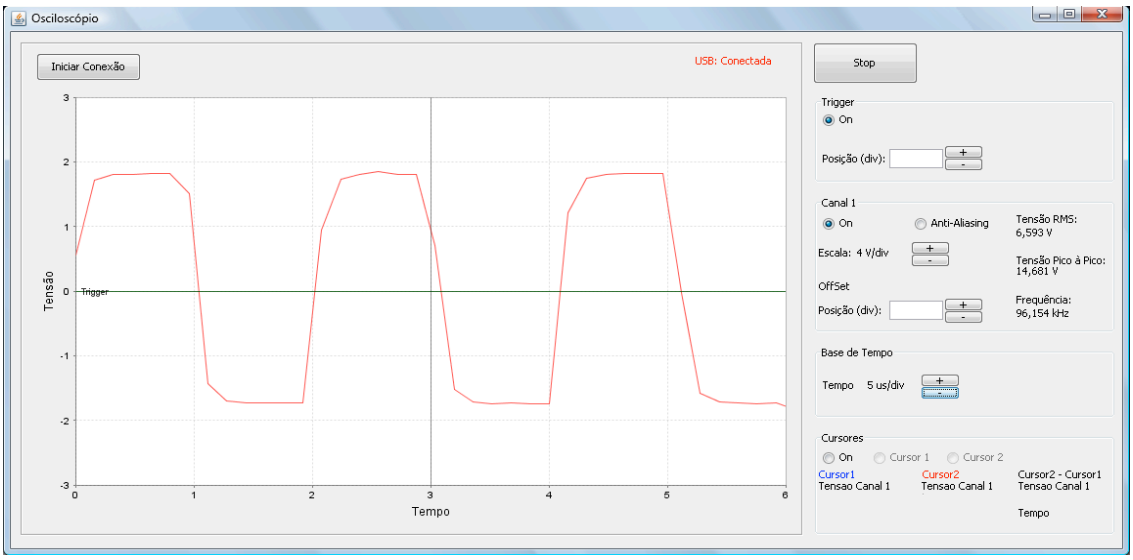


Figura 98 - Osciloscópio projetado [Vpp = 14,681 V; Vrms = 6,593 V; f = 96,154 kHz]

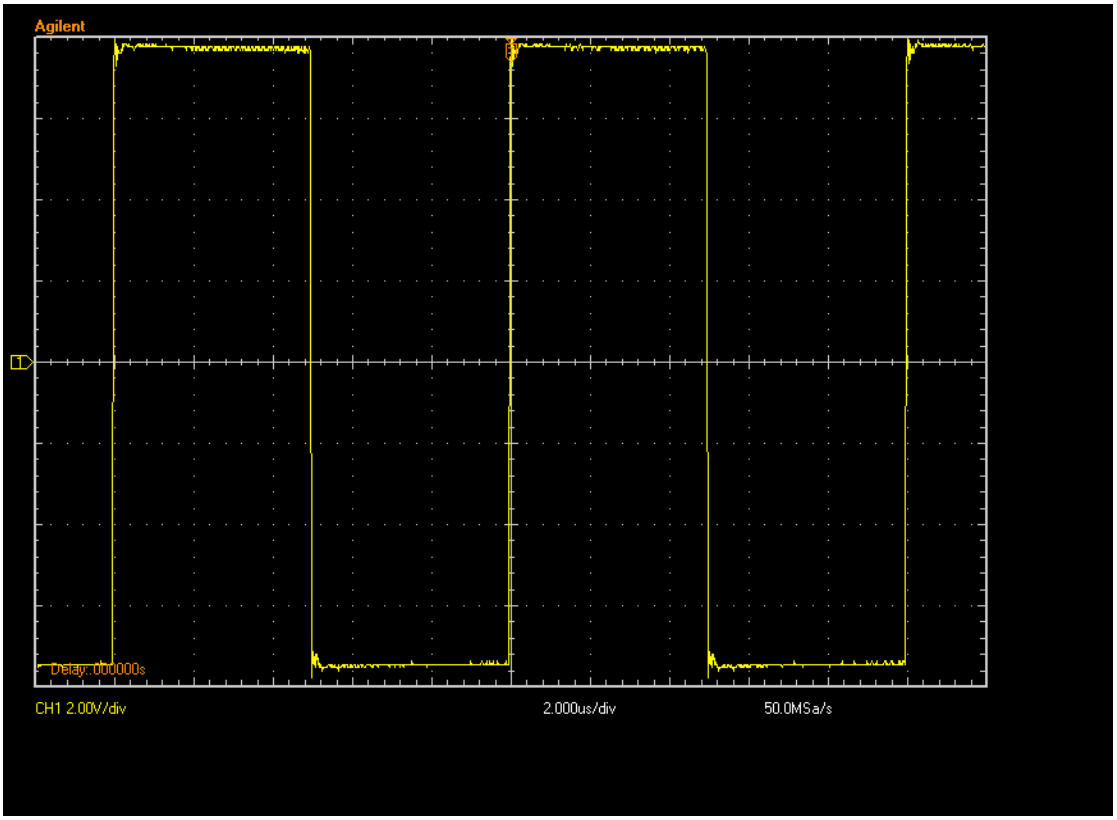


Figura 99 - Osciloscópio comercial [Vpp = 15,7 V; Vrms = 7,57 V; f = 100 kHz]

- Frequência 10 kHz e tensão 15,4 V

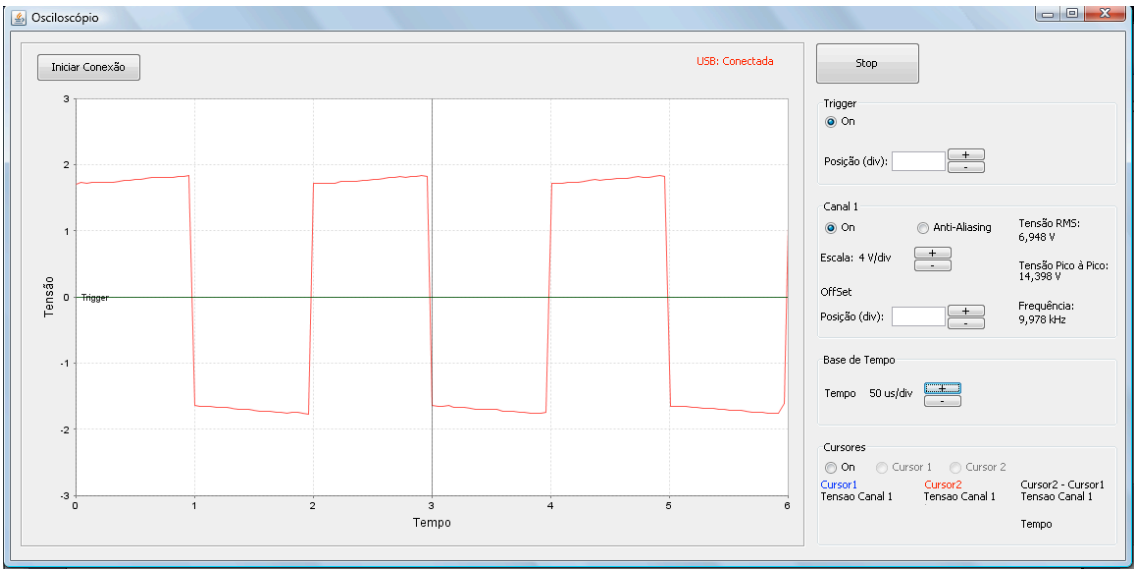


Figura 100 - Osciloscópio projetado [Vpp = 14,398 V; Vrms = 6,948 V; f = 9,978 kHz]

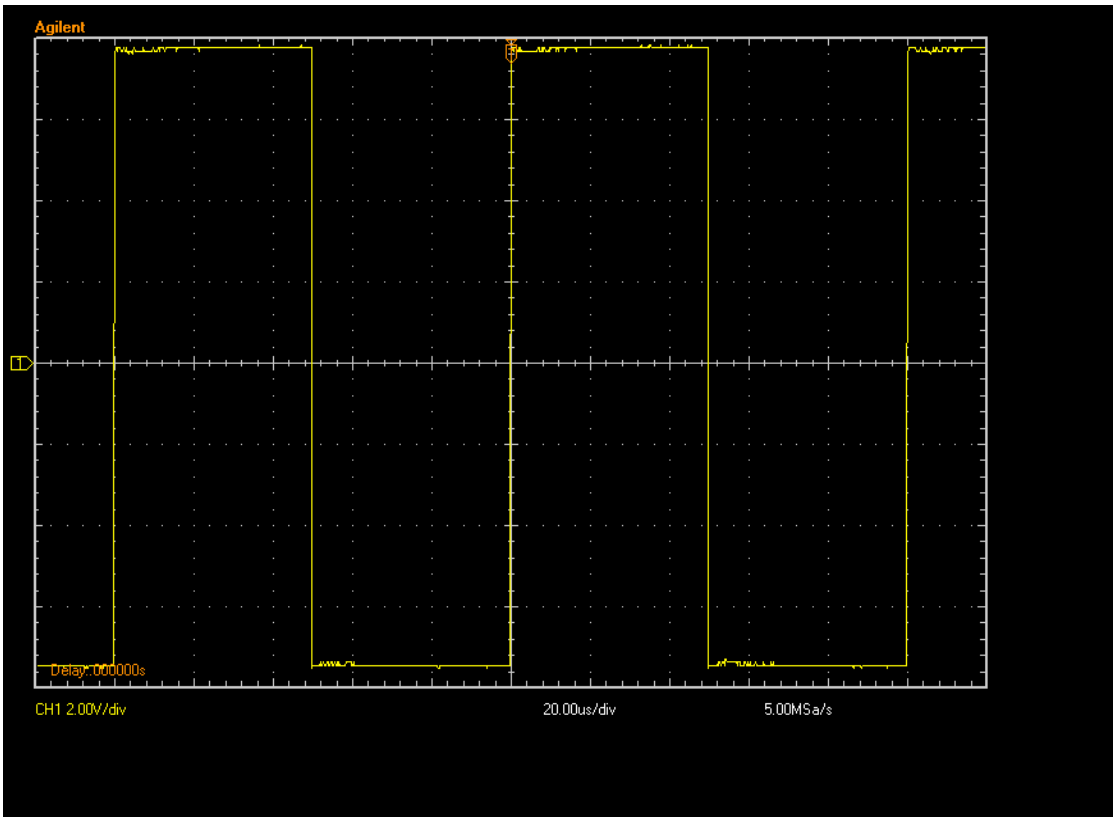


Figura 101 - Osciloscópio comercial [Vpp = 15,4 V; Vrms = 7,59 V; f = 10 kHz]

- Frequência 1 kHz e tensão 15,5 V

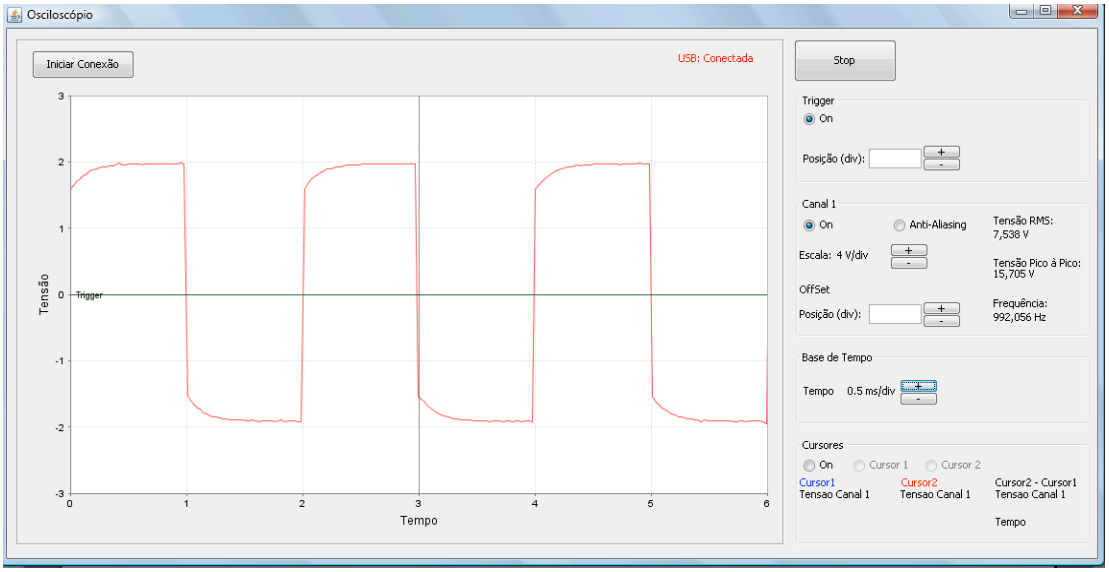


Figura 102 - Osciloscópio projetado [Vpp = 15,705 V; Vrms = 7,538 V; f = 992,056 Hz]

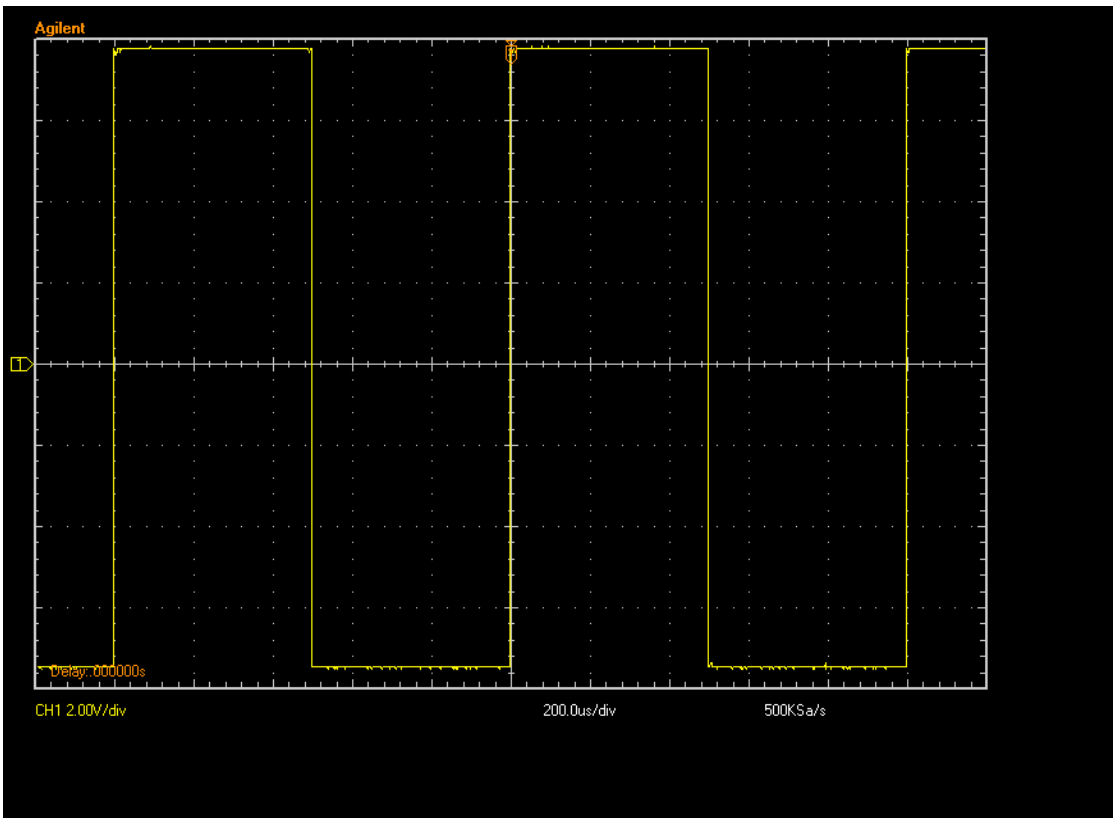


Figura 103 - Osciloscópio comercial [Vpp = 15,5 V; Vrms = 7,6 V; f = 1 kHz]

- Frequência 1,01 Hz e tensão 15,3 V

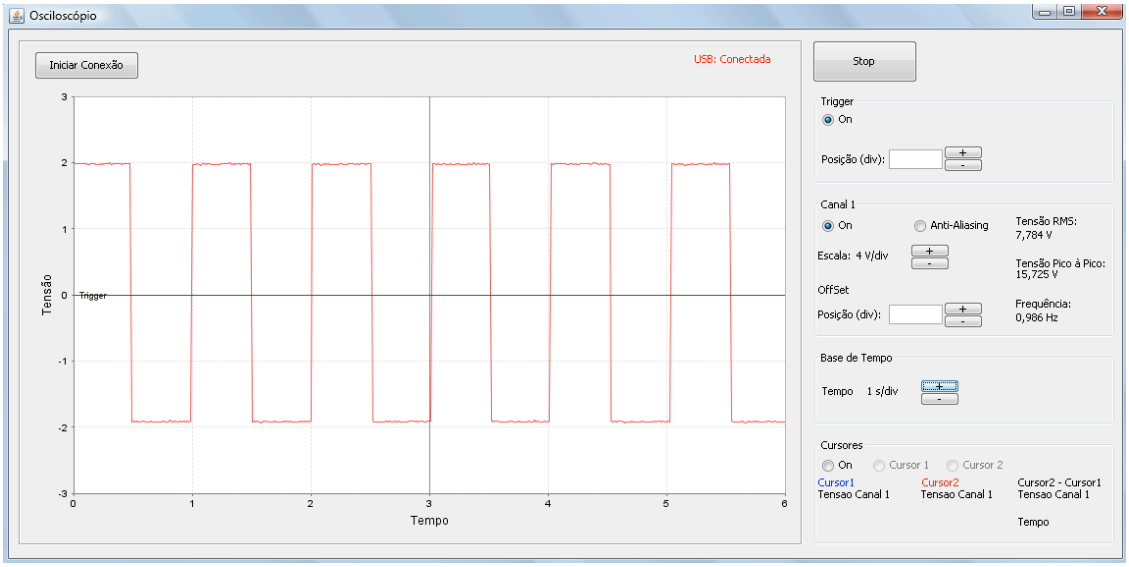


Figura 104 - Osciloscópio projetado [Vpp = 15,725 V; Vrms = 7,784 V; f = 0,986 Hz]

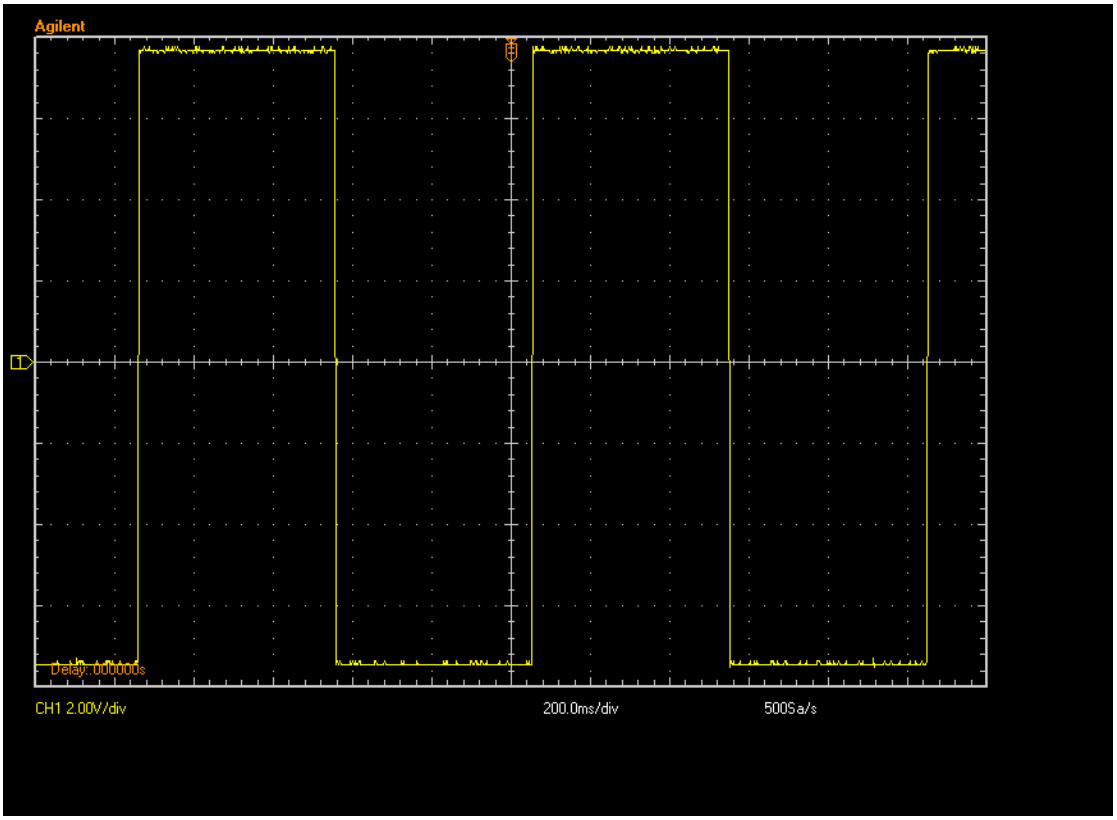


Figura 105 - Osciloscópio comercial [Vpp = 15,3 V; Vrms = 7,55 V; f = 1,01 Hz]

- Frequência 1 Hz e tensão 2,08 V

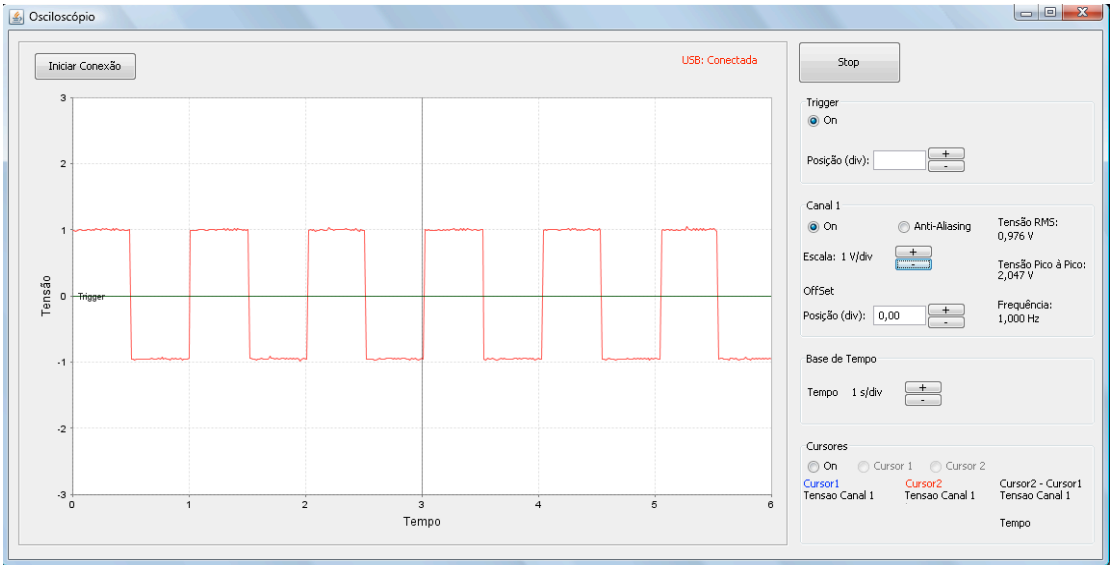


Figura 106 - Osciloscópio projetado [Vpp = 2,047 V; Vrms = 0,976 V; f = 1 Hz]

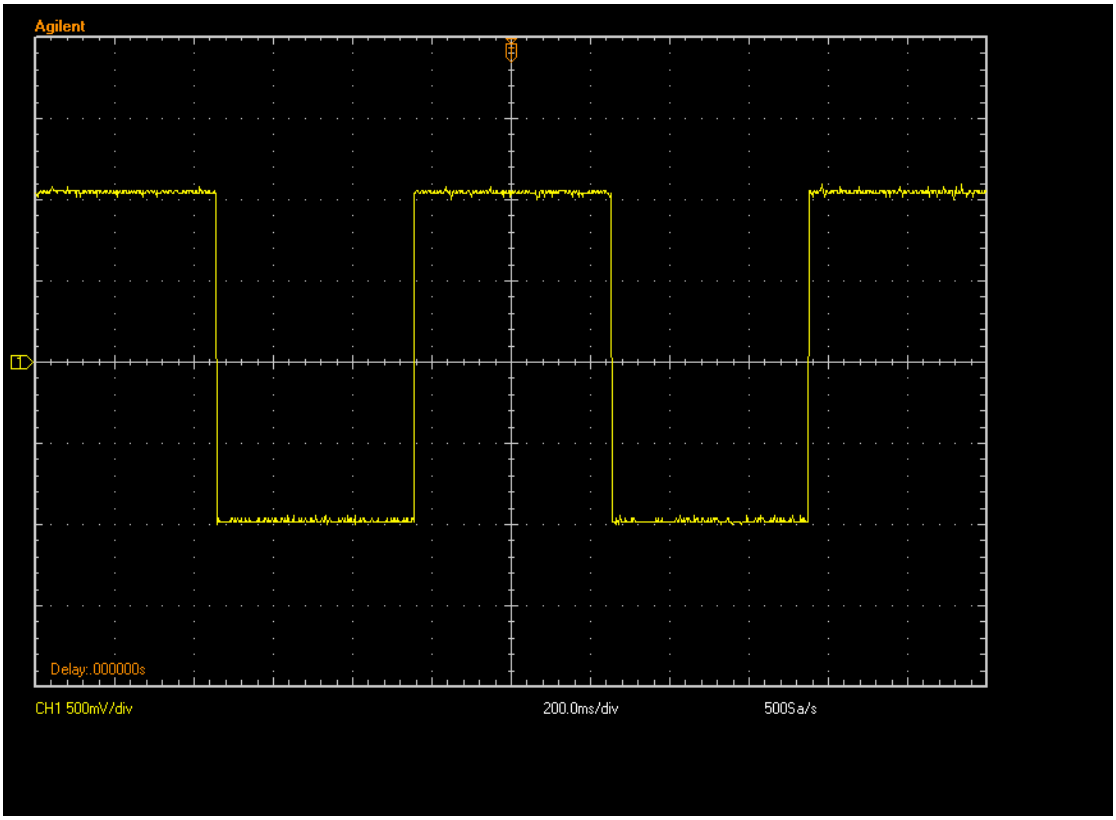


Figura 107 - Osciloscópio comercial [Vpp = 2,08 V; Vrms = 1,02 V; f = 1 Hz]

- Frequência 1 kHz e tensão 2,08 V

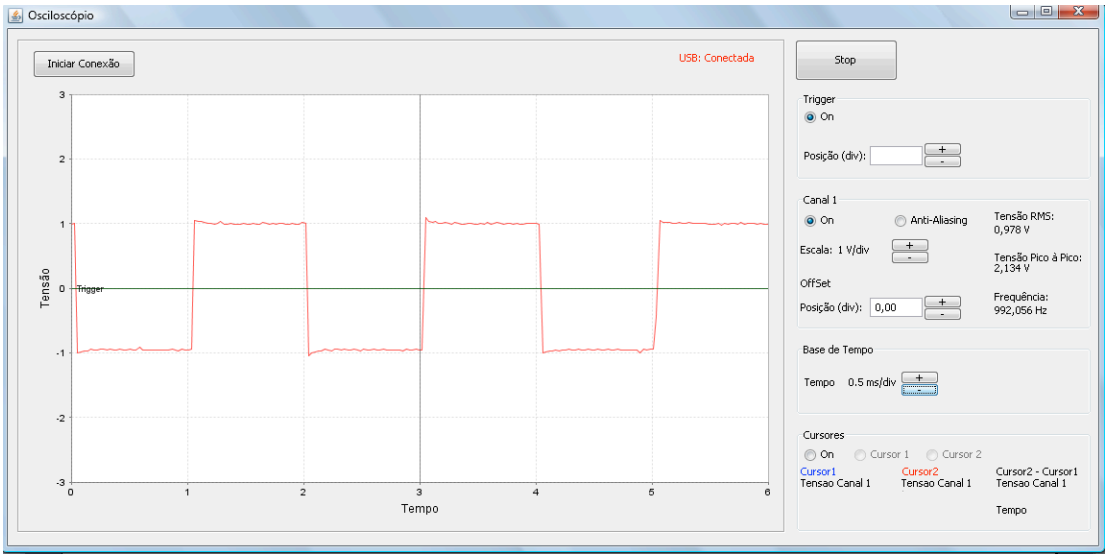


Figura 108 - Osciloscópio projetado [Vpp = 2,134 V; Vrms = 0,978 V; f = 992,056 Hz]

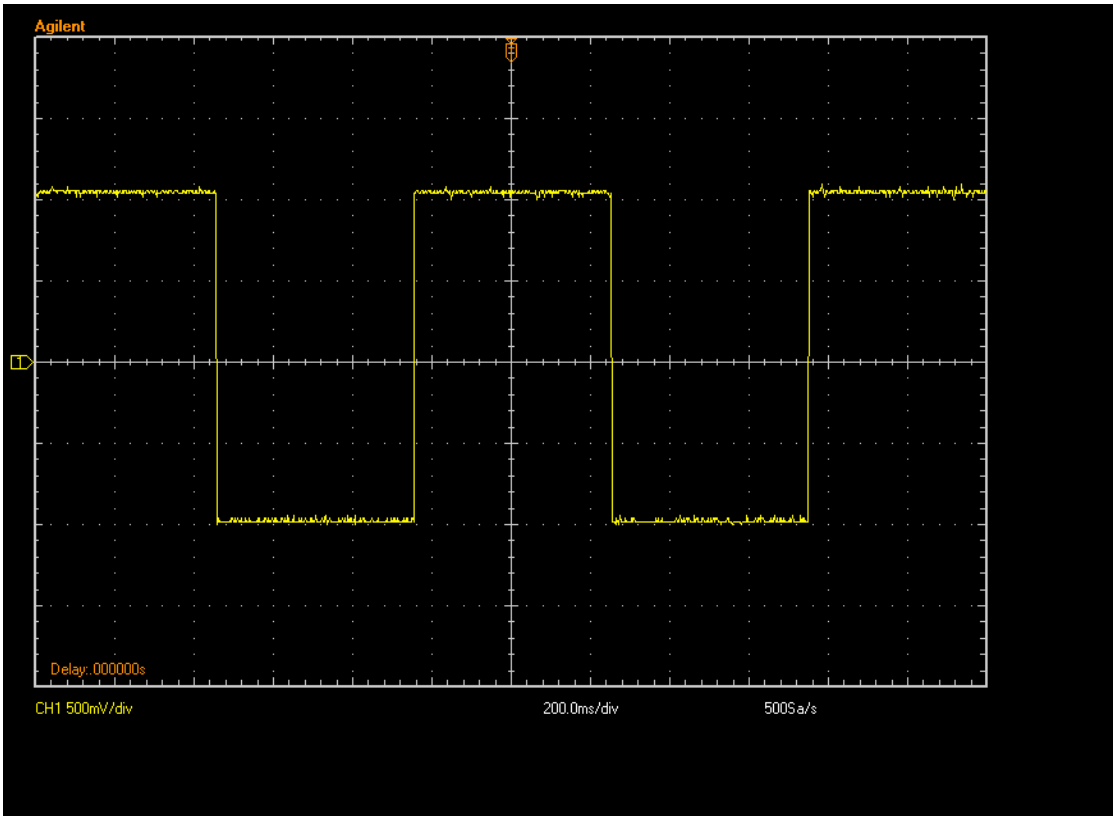


Figura 109 - Osciloscópio comercial [Vpp = 2,08 V; Vrms = 1,02 V; f = 1 kHz]

- Frequência 10 kHz e tensão 2,08 V

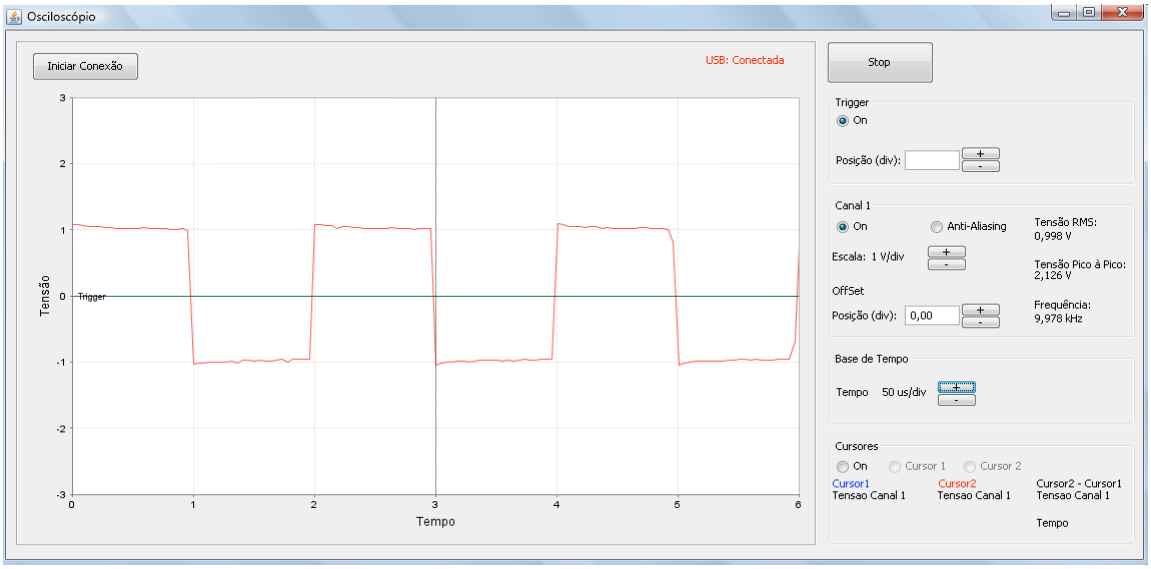


Figura 110 - Osciloscópio projetado [Vpp = 2,126 V; Vrms = 0,998 V; f = 9,978 kHz]

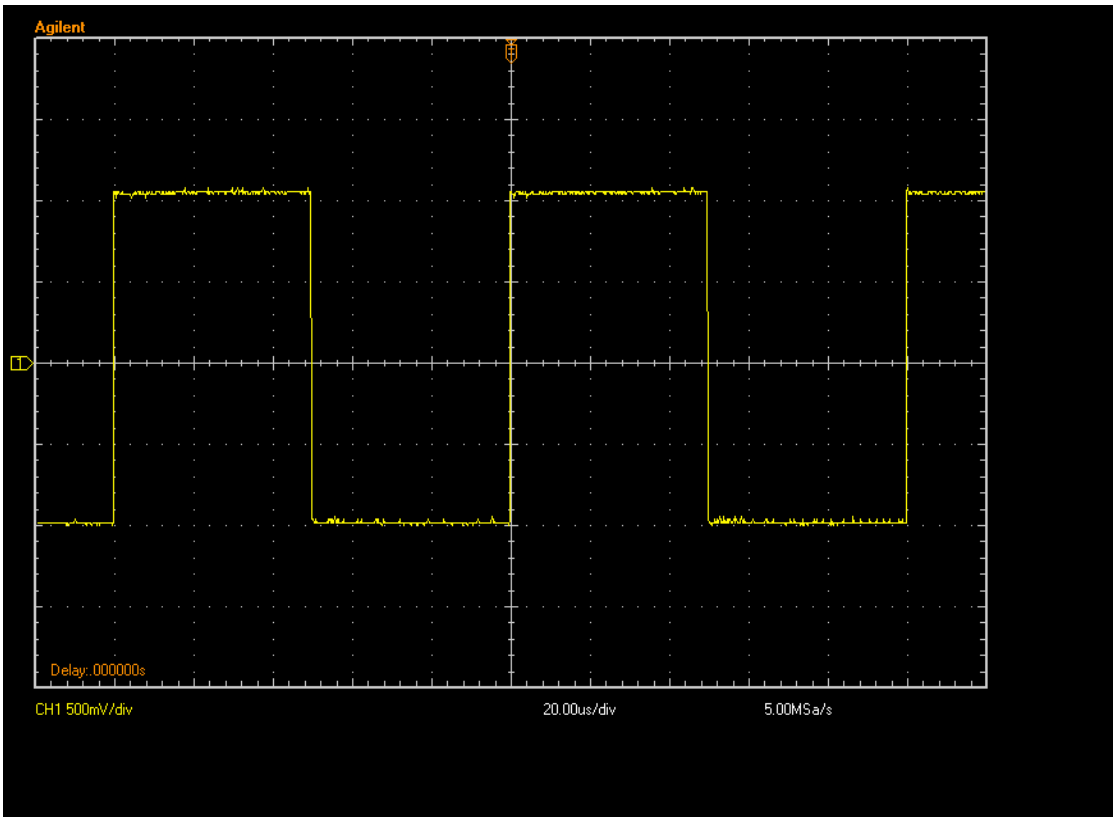


Figura 111 - Osciloscópio comercial [Vpp = 2,08 V; Vrms = 1,02 V; f = 10 kHz]

- Frequência 100 kHz e tensão 2,14 V

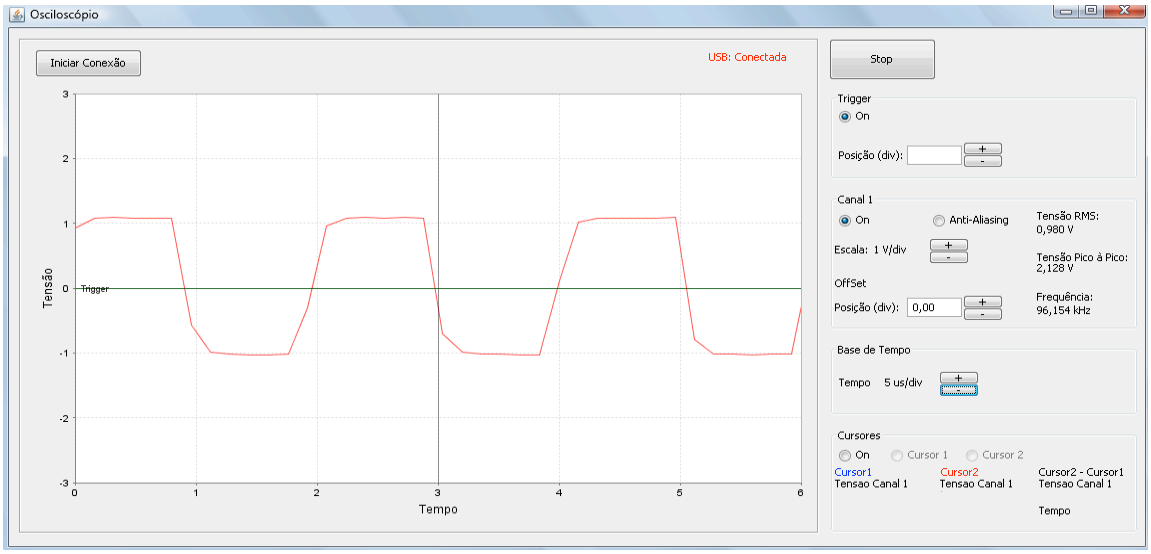


Figura 112 - Osciloscópio projetado [Vpp = 2,128 V; Vrms = 0,980 V; f = 96,154 kHz]

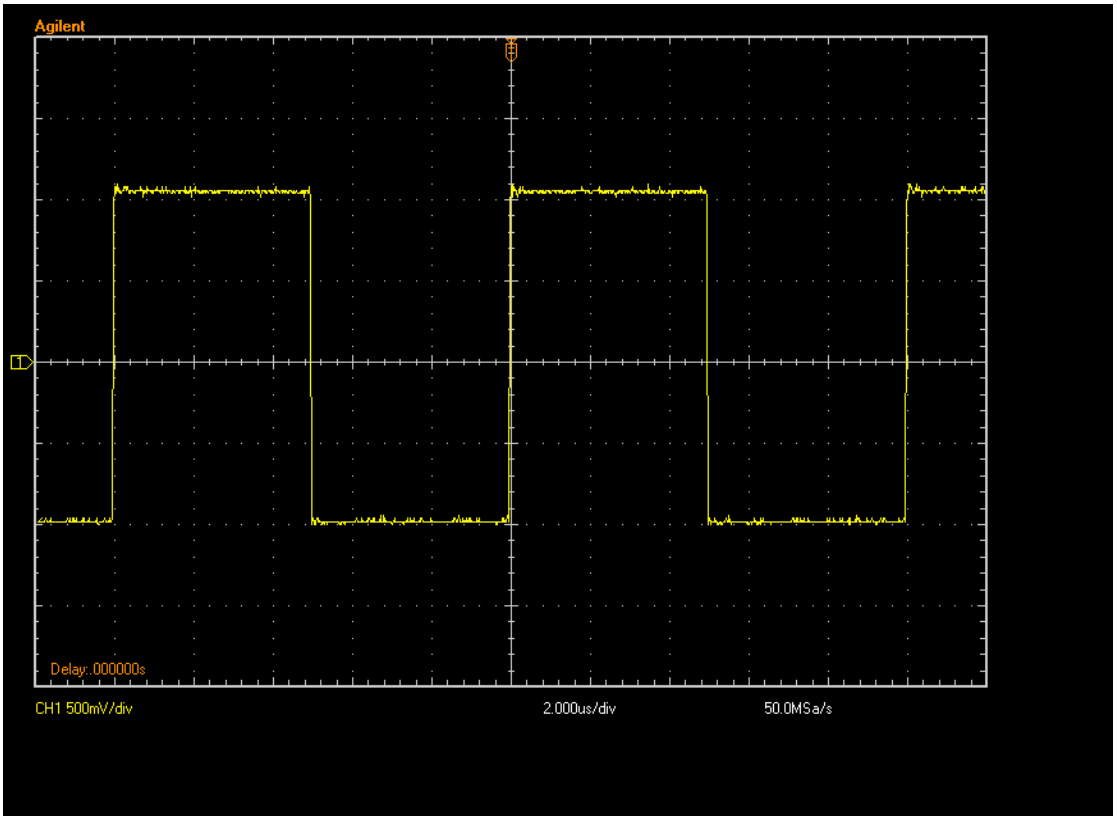


Figura 113 - Osciloscópio comercial [Vpp = 2,14 V; Vrms = 1,01 V; f = 100 kHz]

- Frequência 100 kHz e tensão 2,04 V

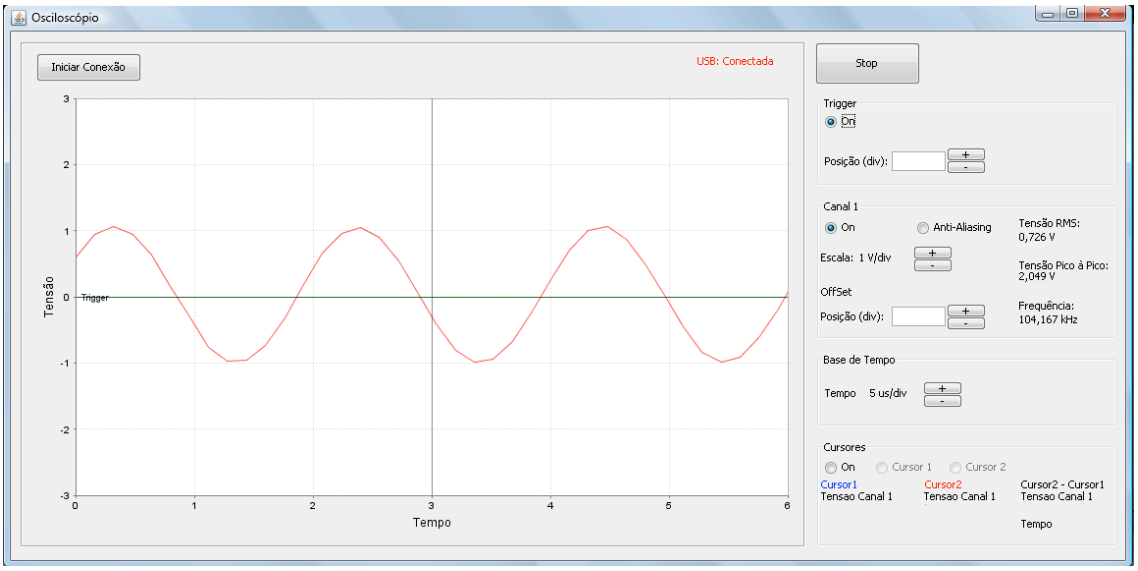


Figura 114 - Osciloscópio projetado [Vpp = 2,049 V; Vrms = 0,726 V; f = 104,167 kHz]

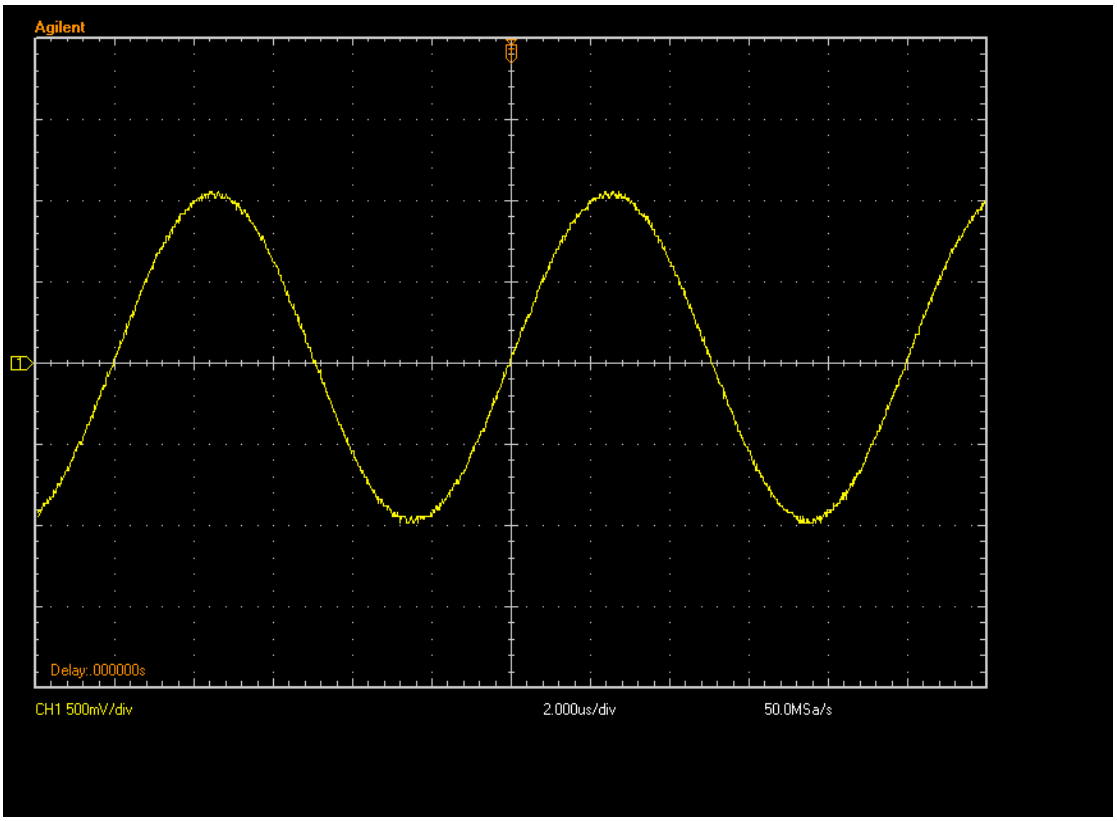


Figura 115 - Osciloscópio comercial [Vpp = 2,04 V; Vrms = 0,696 V; f = 100 kHz]

- Frequência 10 kHz e tensão 2,06 V

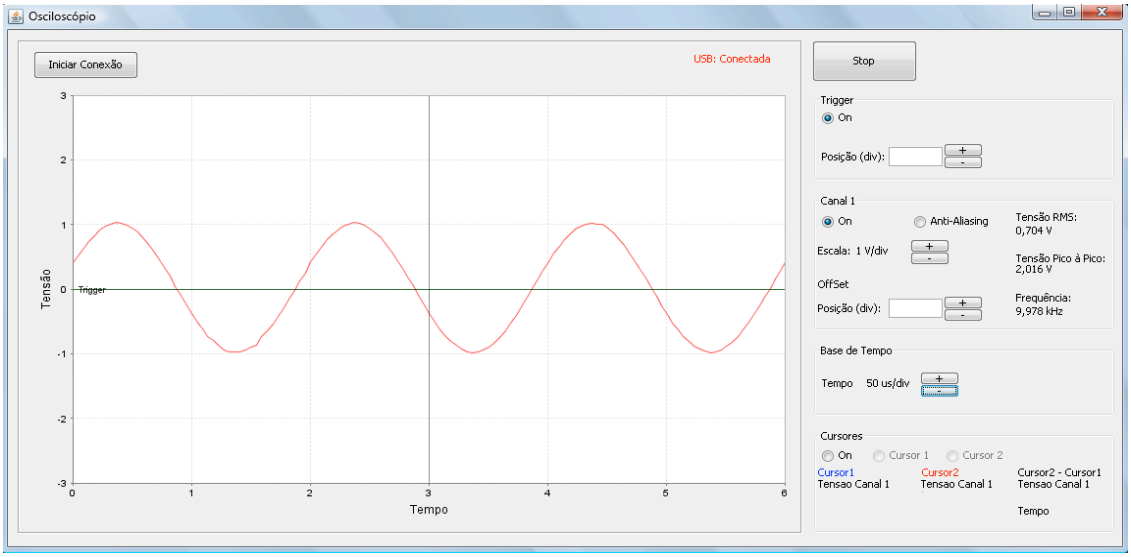


Figura 116 - Osciloscópio projetado [Vpp = 2,016 V; Vrms = 0,704 V; f = 9,978 kHz]

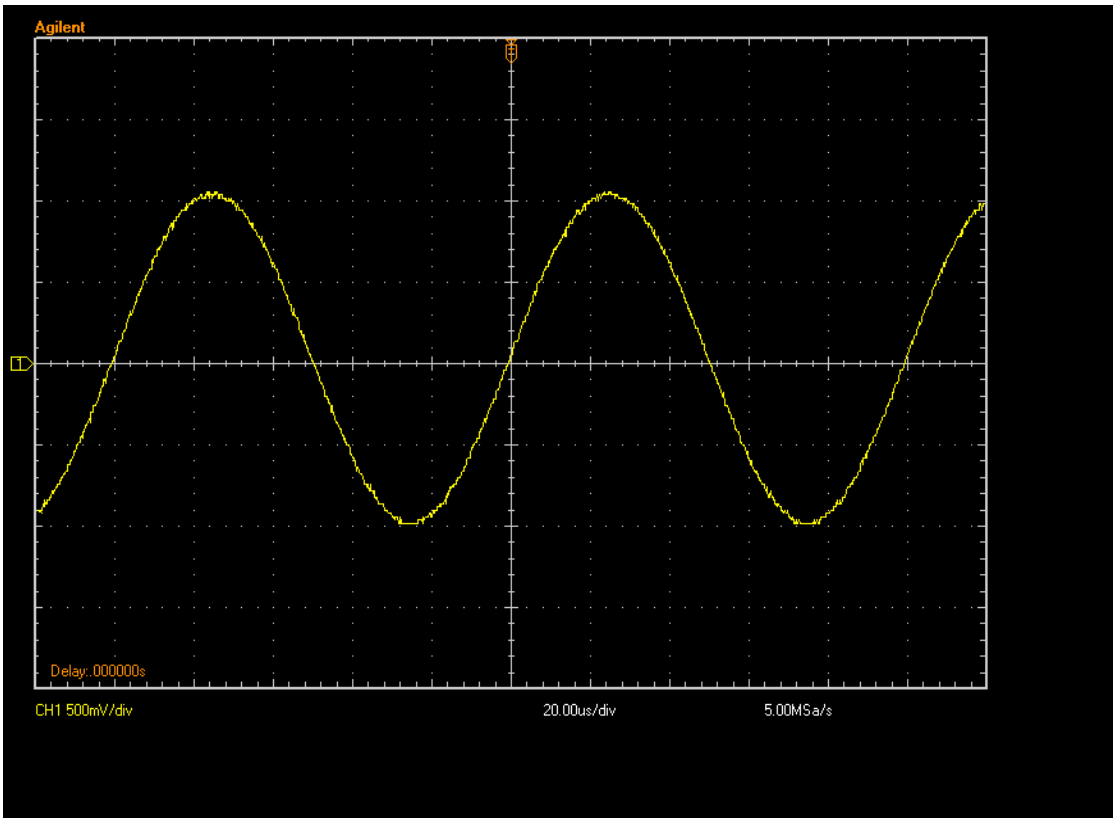


Figura 117 - Osciloscópio comercial [Vpp = 2,06 V; Vrms = 0,701 V; f = 10 kHz]

- Frequência 1 kHz e tensão 2,04 V

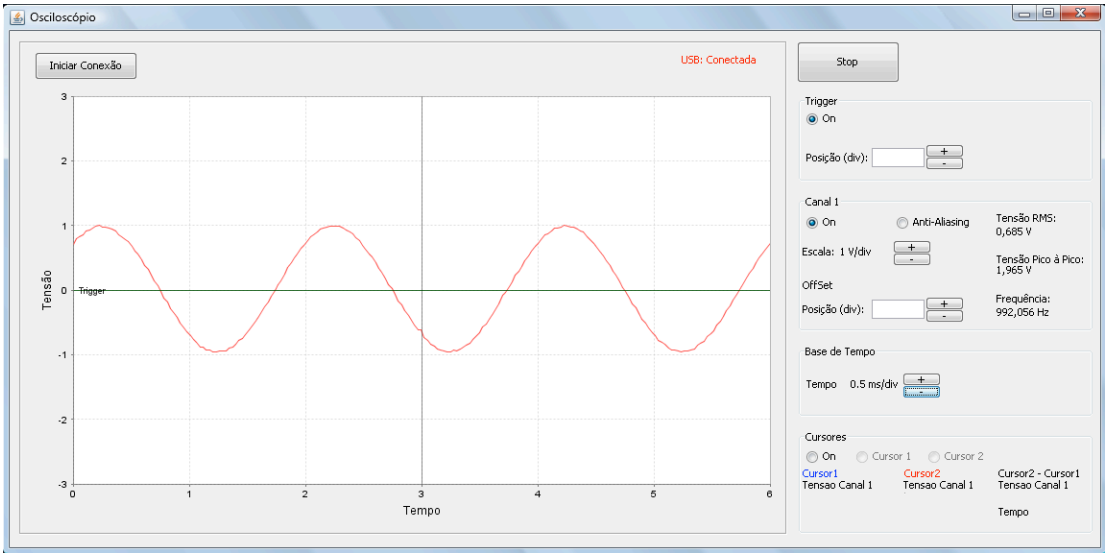


Figura 118 - Osciloscópio projetado [Vpp = 1,965 V; Vrms = 0,685 V; f = 992,056 Hz]

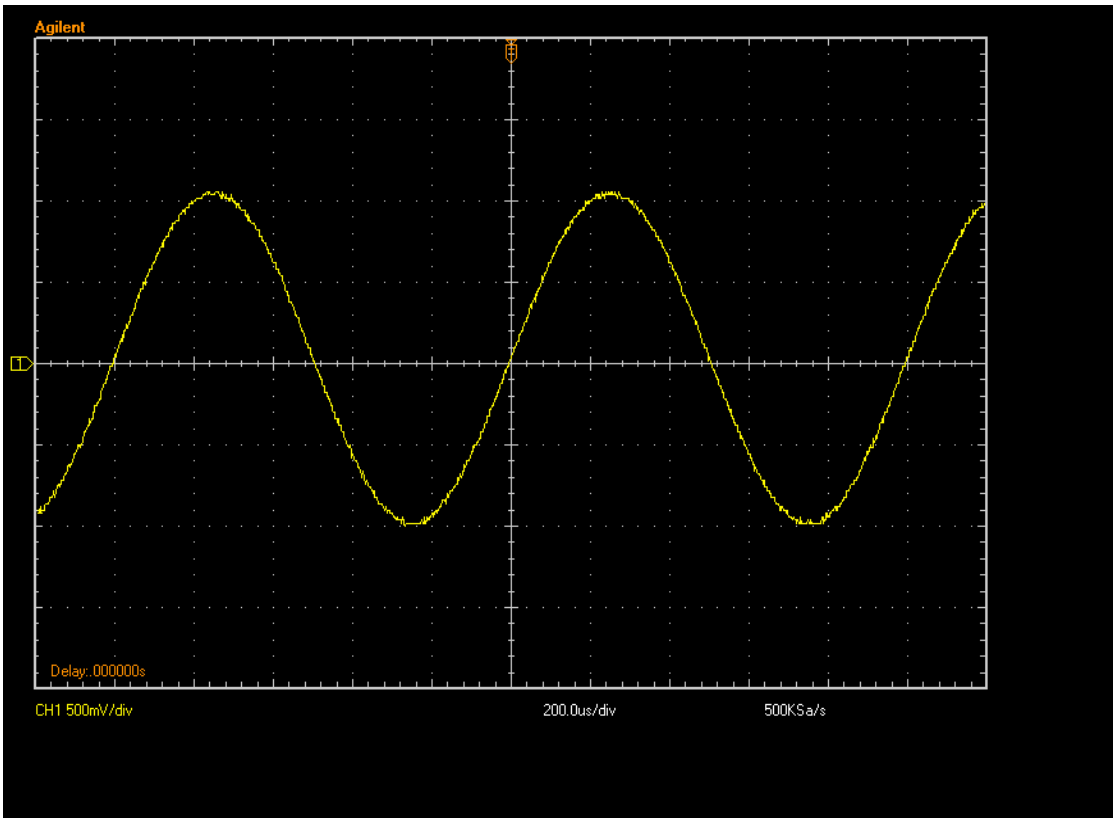


Figura 119 - Osciloscópio comercial [Vpp = 2,04 V; Vrms = 0,702 V; f = 1 kHz]

- Frequência 1,01 Hz e tensão 2,04 V

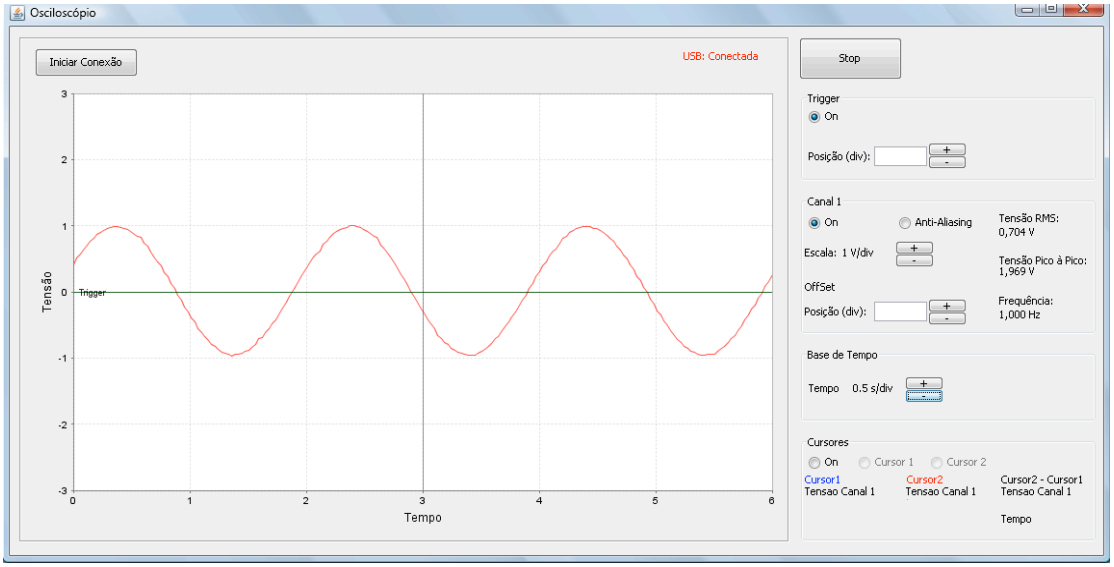


Figura 120 - Osciloscópio projetado [Vpp = 1,969 V; Vrms = 0,704 V; f = 1 Hz]

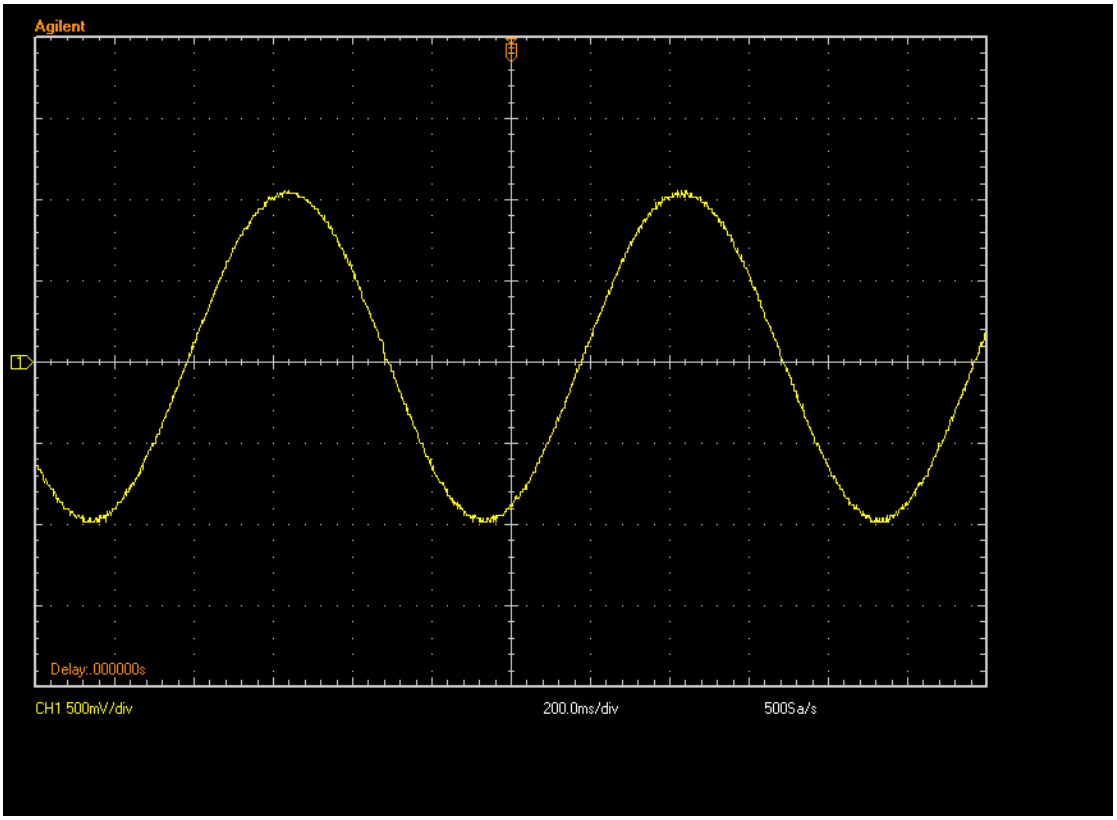


Figura 121 - Osciloscópio comercial [Vpp = 2,04 V; Vrms = 0,718 V; f = 1,01 Hz]

- Frequência 1 Hz e tensão 1,05 V

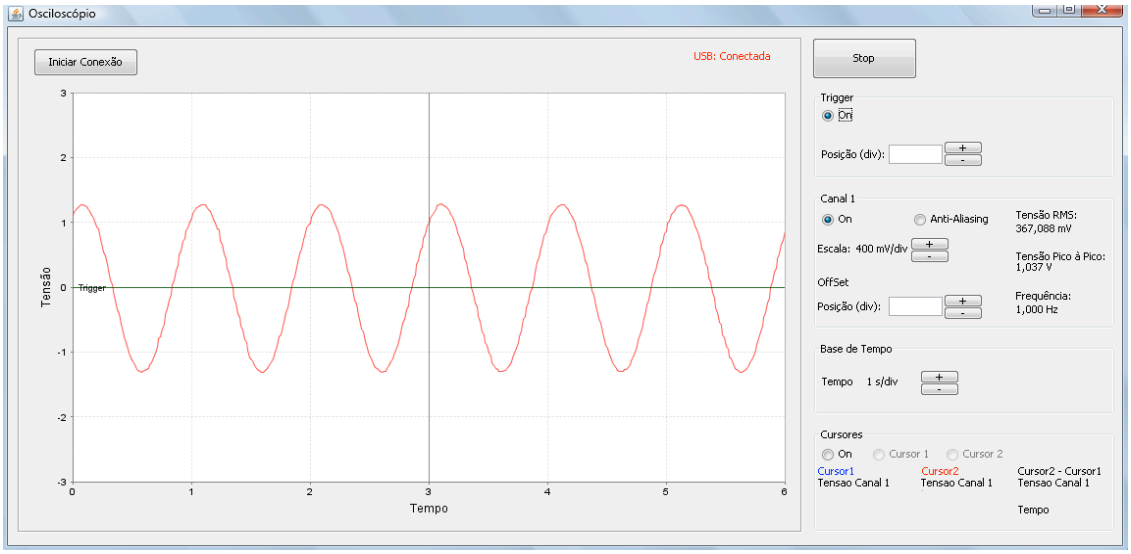


Figura 122 - Osciloscópio projetado [Vpp = 1,037 V; Vrms = 367,088 mV; f = 1 Hz]

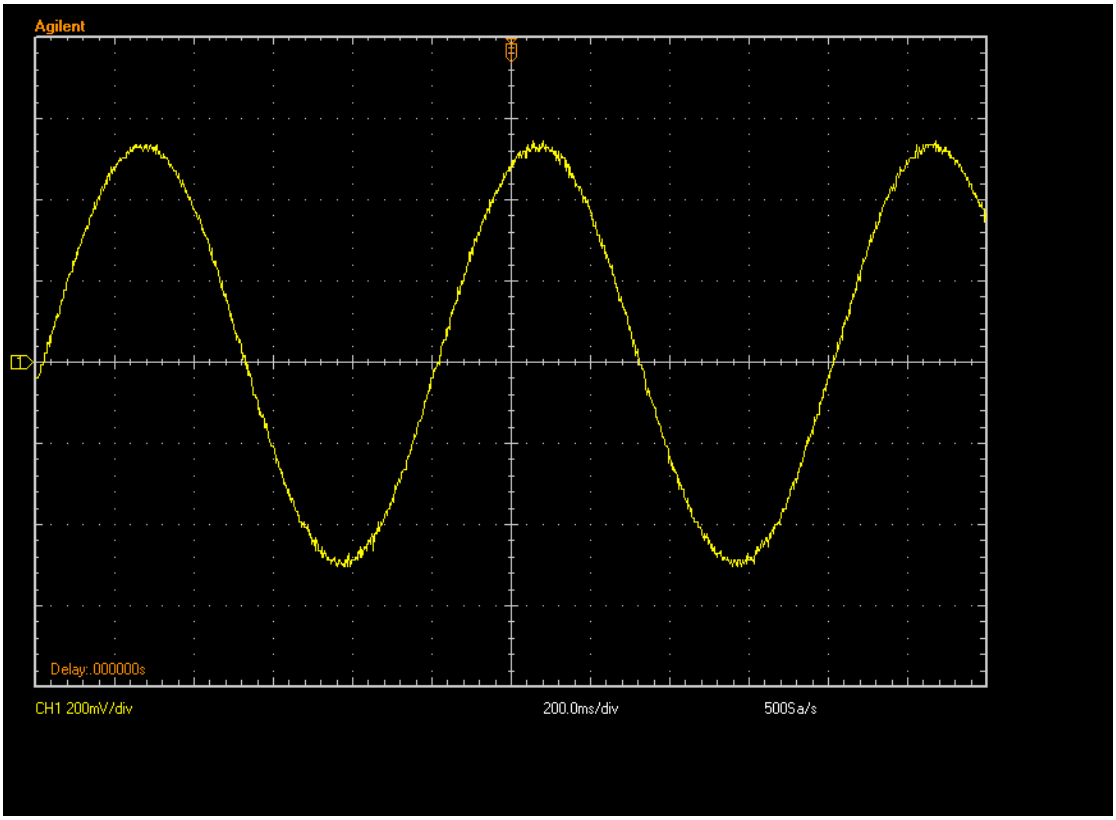


Figura 123 - Osciloscópio comercial [Vpp = 1,05 V; Vrms = 369 mV; f = 1 Hz]

- Frequência 1 kHz e tensão 1,09 V

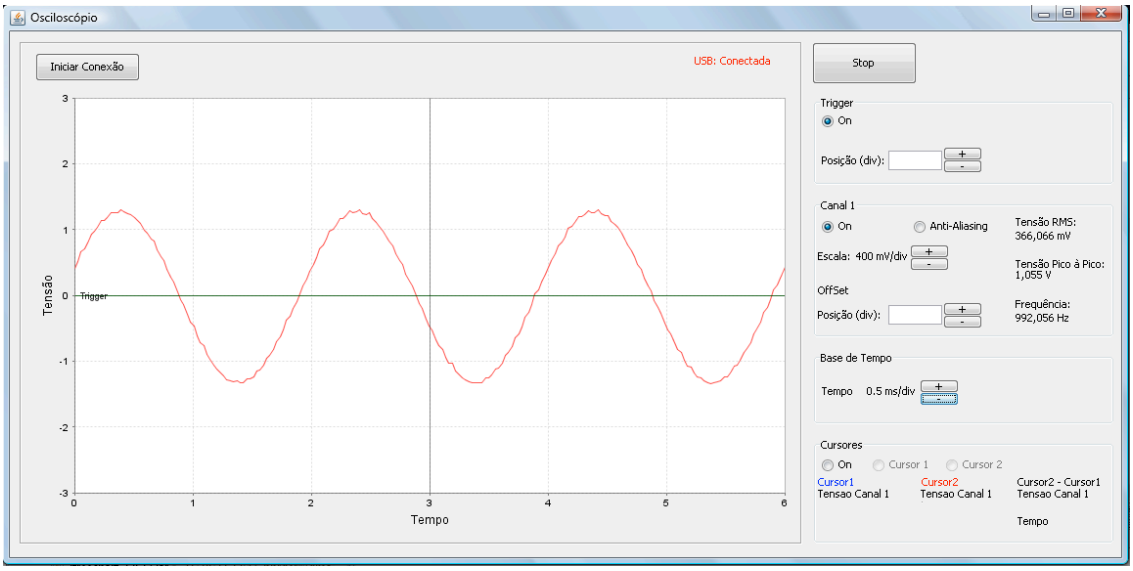


Figura 124 - Osciloscópio projetado[Vpp = 1,055V; Vrms = 366,066mV; f = 992,056Hz]

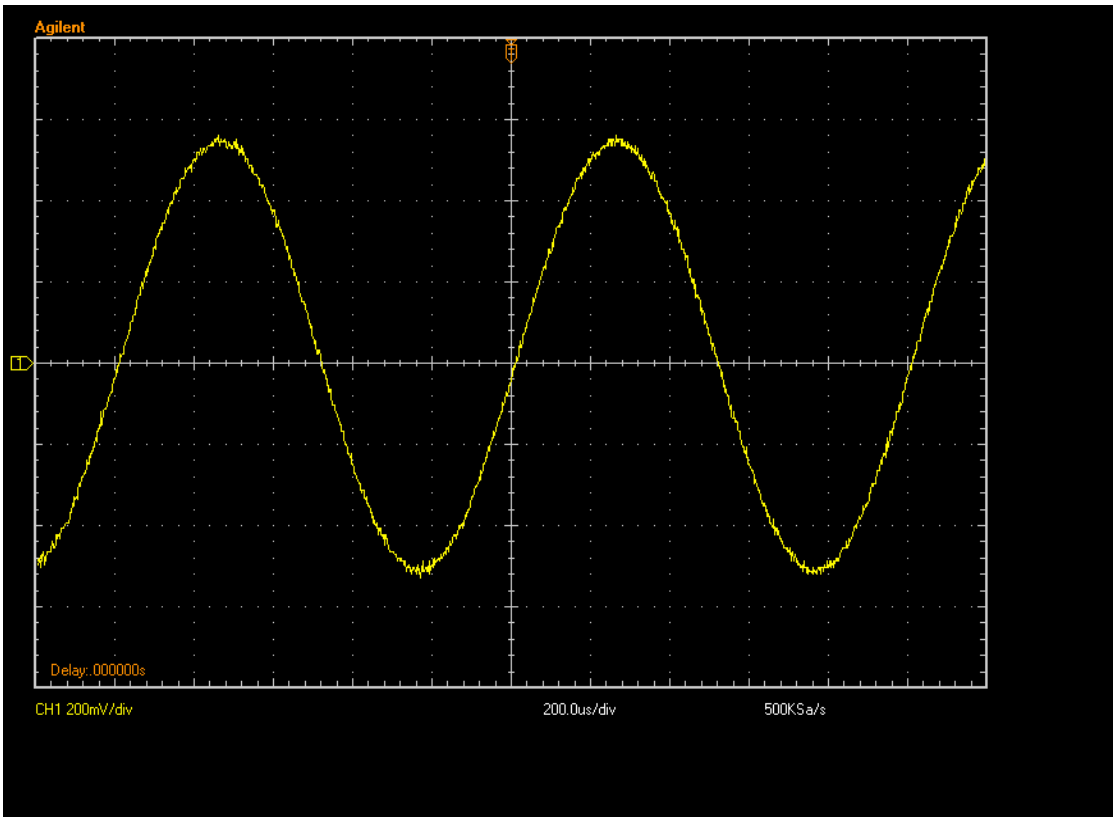


Figura 125 - Osciloscópio comercial [Vpp = 1,09 V; Vrms = 366 mV; f = 1 kHz]

- Frequência 10 kHz e tensão 1,08 V

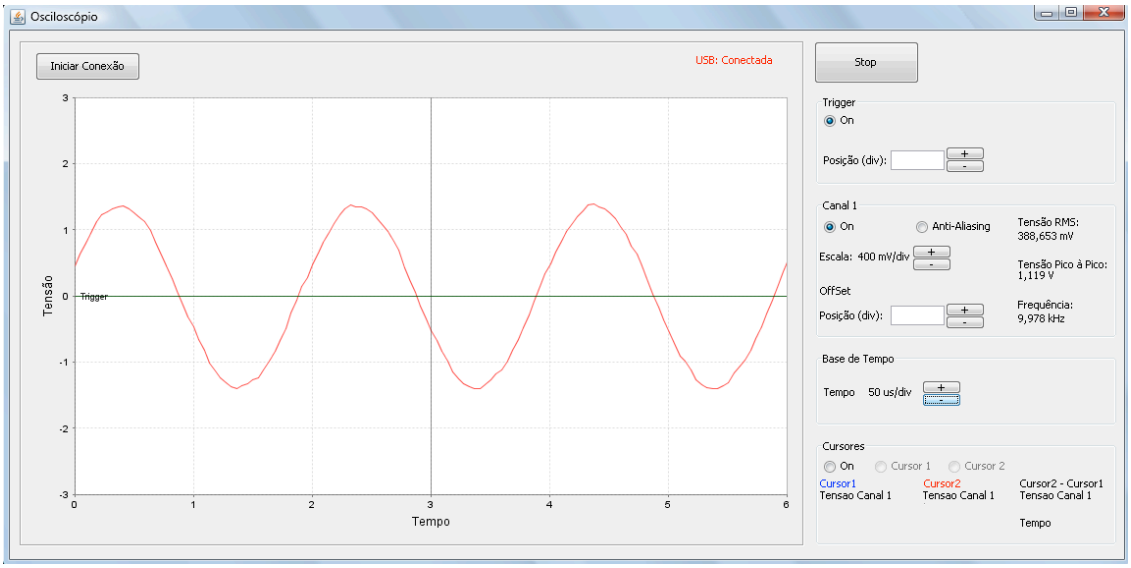


Figura 126 - Osciloscópio projetado [Vpp = 1,119 V; Vrms = 386,653 mV; f = 9,978 kHz]

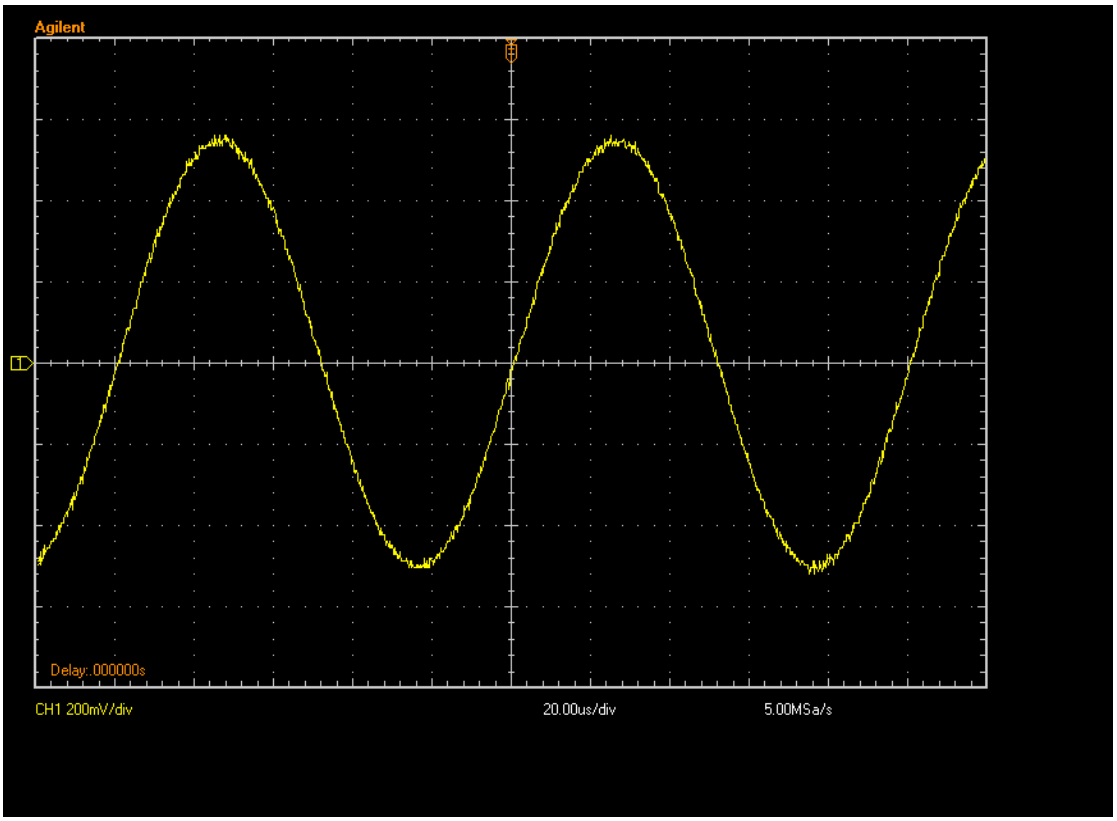


Figura 127 - Osciloscópio comercial [Vpp = 1,08 V; Vrms = 364 mV; f = 10 kHz]

- Frequência 100 kHz e tensão 1,1 V

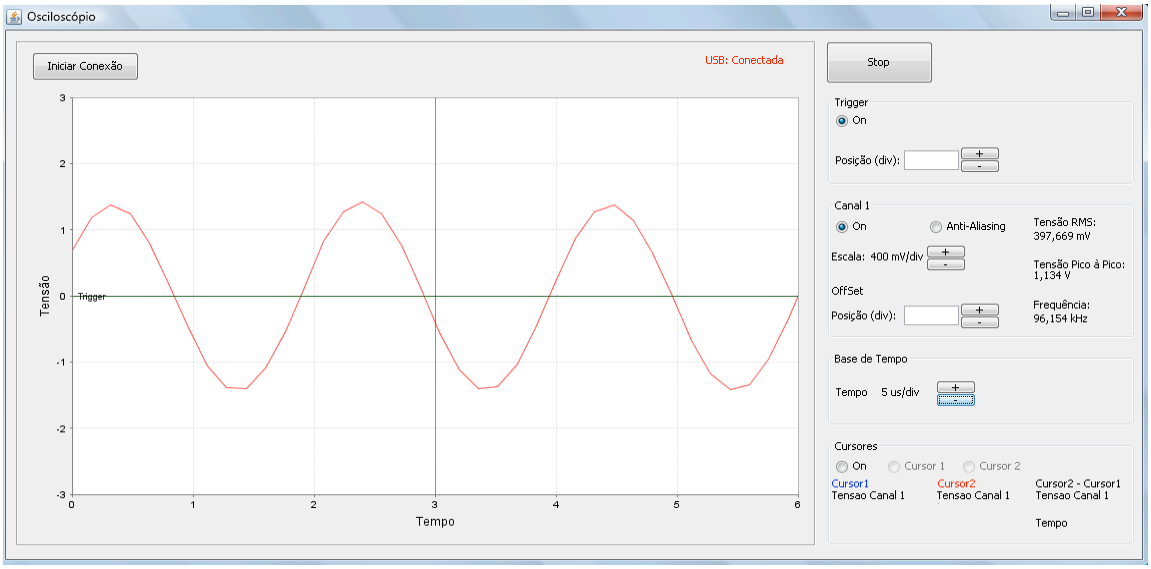


Figura 128 - Osciloscópio projetado [Vpp = 1,134 V; Vrms = 397,669 mV; f = 96,154 kHz]

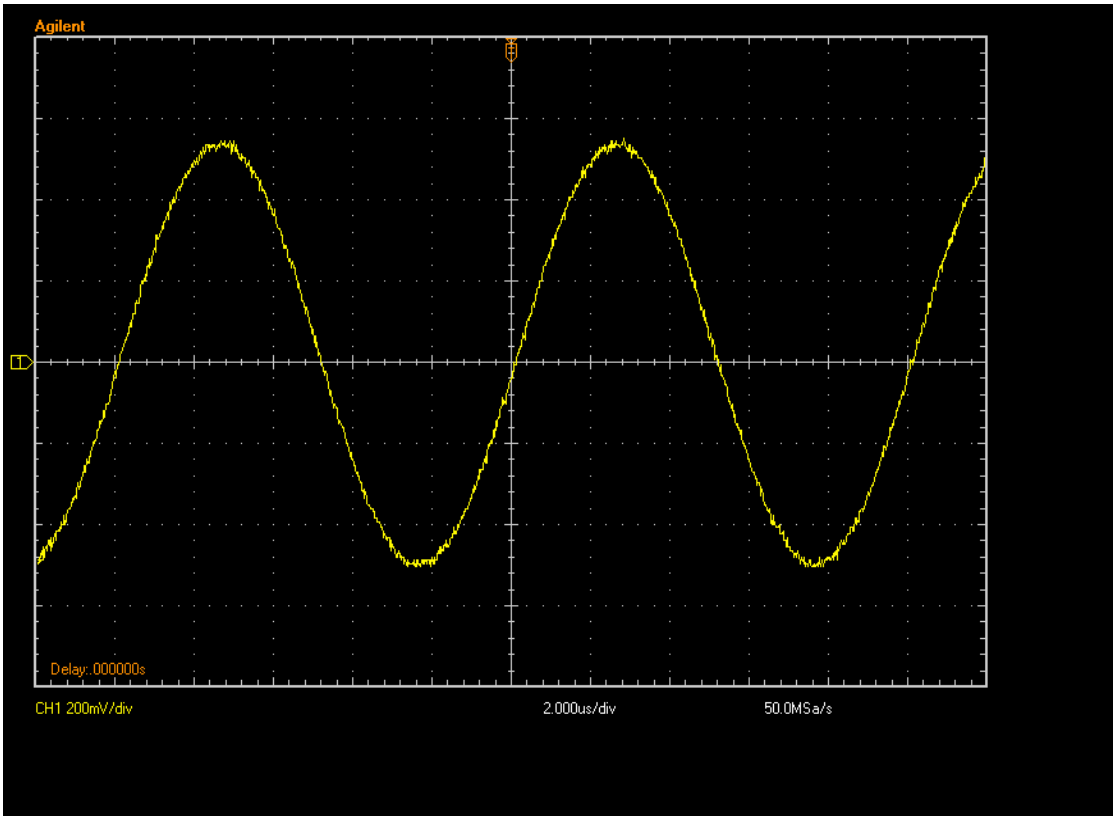


Figura 129 - Osciloscópio comercial [Vpp = 1,07 V; Vrms = 359 mV; f = 101 kHz]

- Frequência 100 kHz e tensão 1,1 V

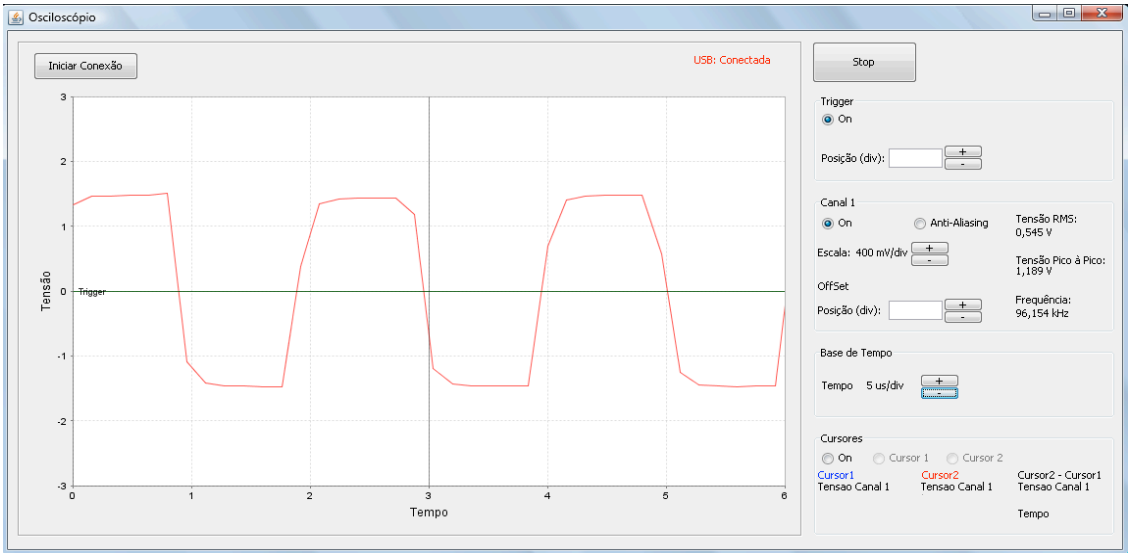


Figura 130 - Osciloscópio projetado [Vpp = 1,189 V; Vrms = 0,545 V; f = 96,154 kHz]

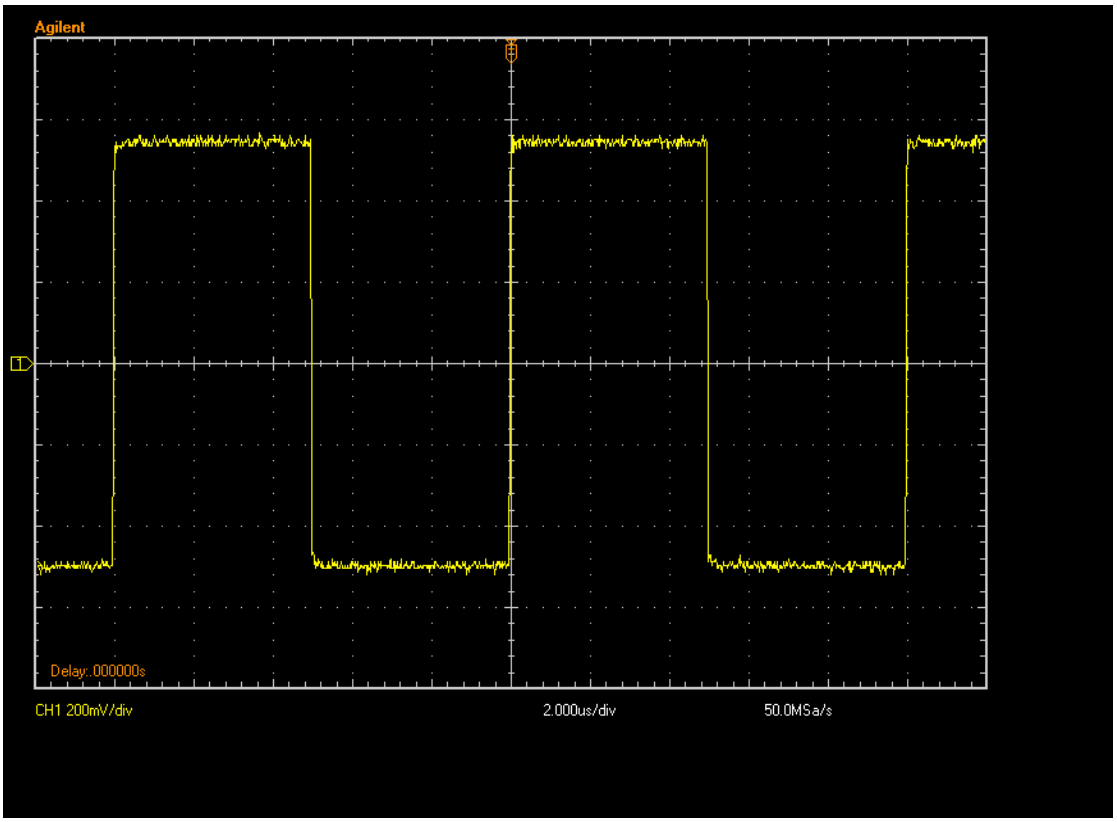


Figura 131 - Osciloscópio comercial [Vpp = 1,1 V; Vrms = 0,520 V; f = 100 kHz]

- Frequência 10 kHz e tensão 1,1 V

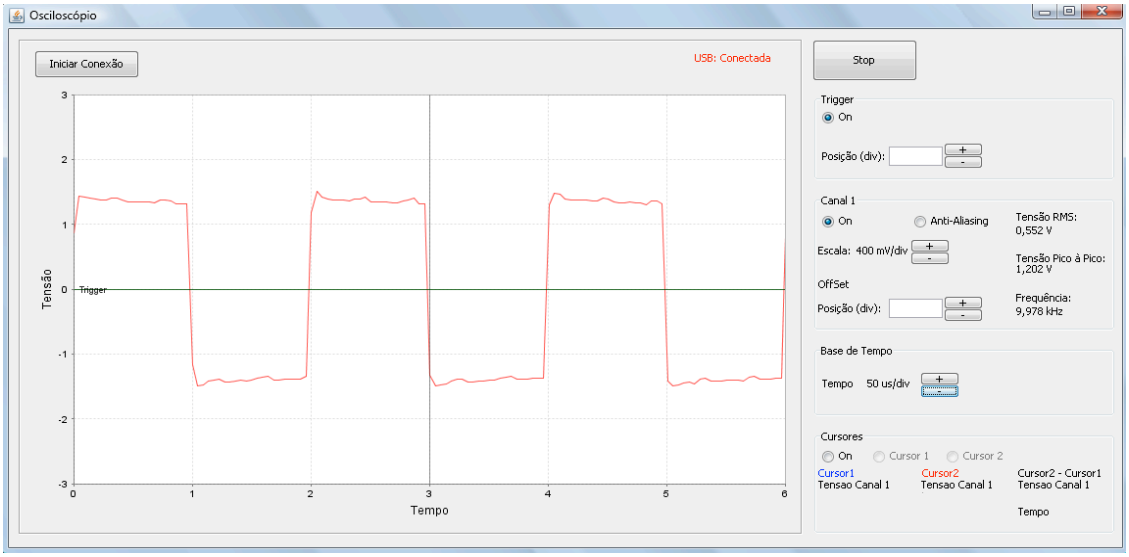


Figura 132 - Osciloscópio projetado [Vpp = 1,202 V; Vrms = 0,552 V; f = 9,978 kHz]

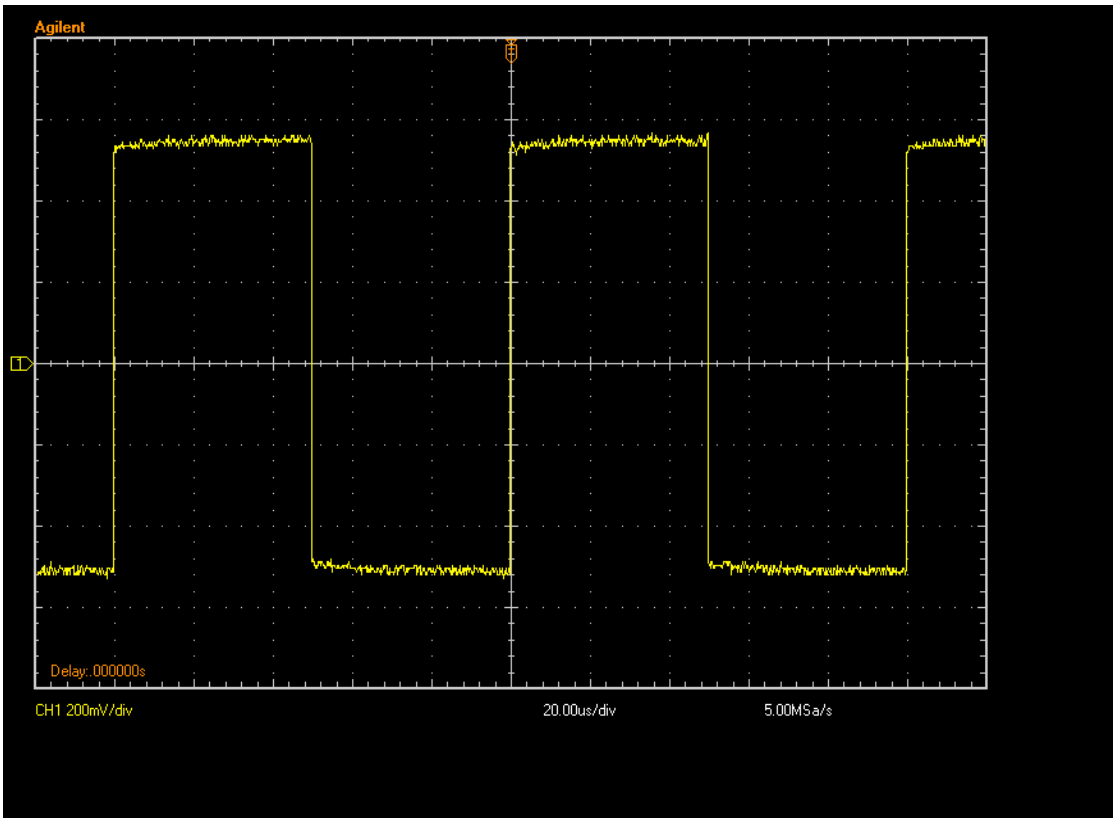


Figura 133 - Osciloscópio comercial [Vpp = 1,1 V; Vrms = 0,526 V; f = 10 kHz]

- Frequência 1 kHz e tensão 1,1 V

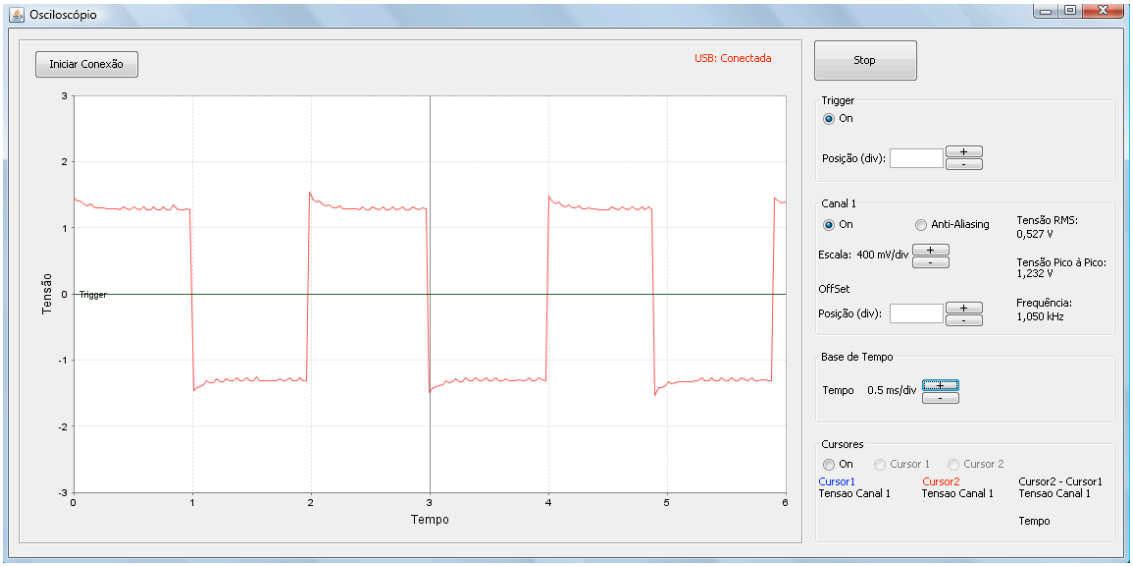


Figura 134 - Osciloscópio projetado [Vpp = 1,232 V; Vrms = 0,527 V; f = 1,05 kHz]

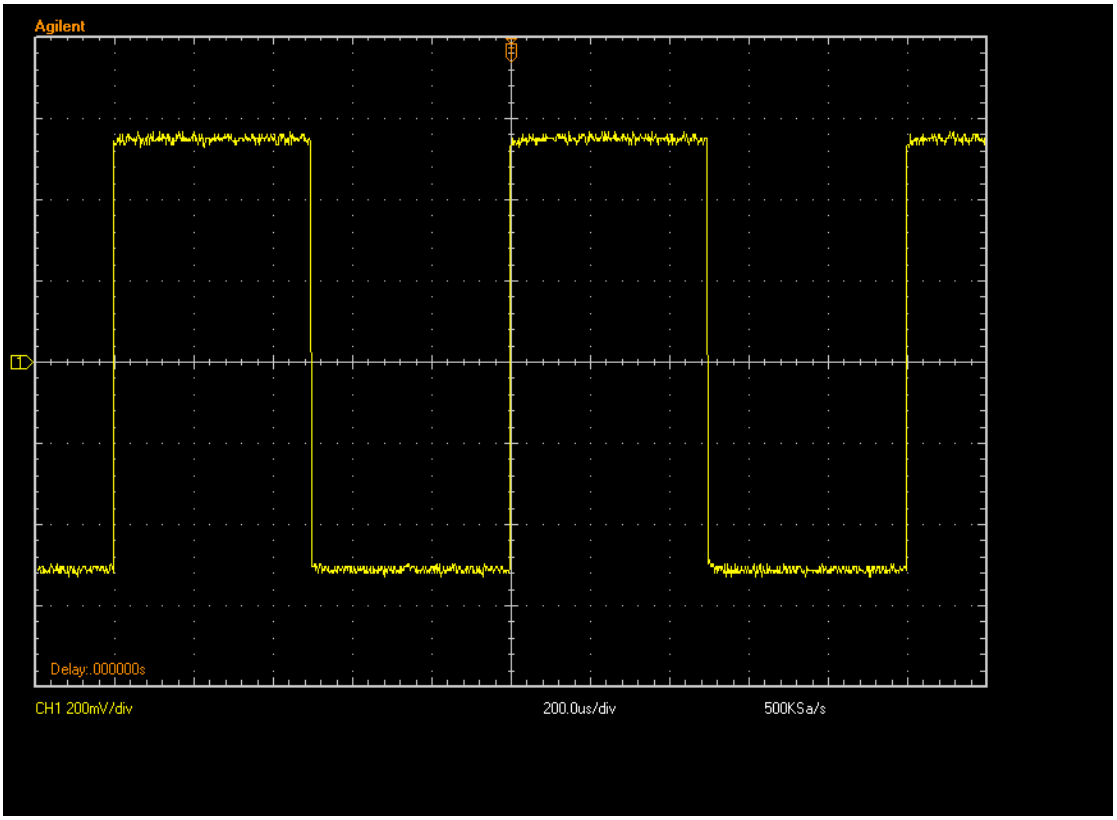


Figura 135 - Osciloscópio comercial [Vpp = 1,1 V; Vrms = 0,530 V; f = 1 kHz]

- Frequência 1 Hz e tensão 1,09 V

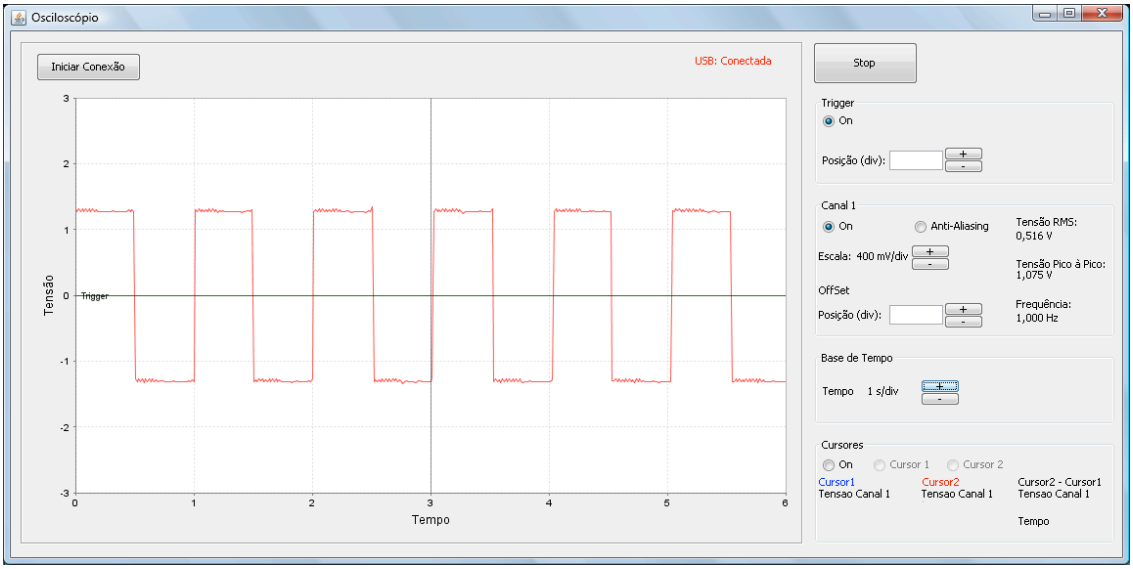


Figura 136 - Osciloscópio projetado [Vpp = 1,075 V; Vrms = 0,516 V; f = 1 Hz]

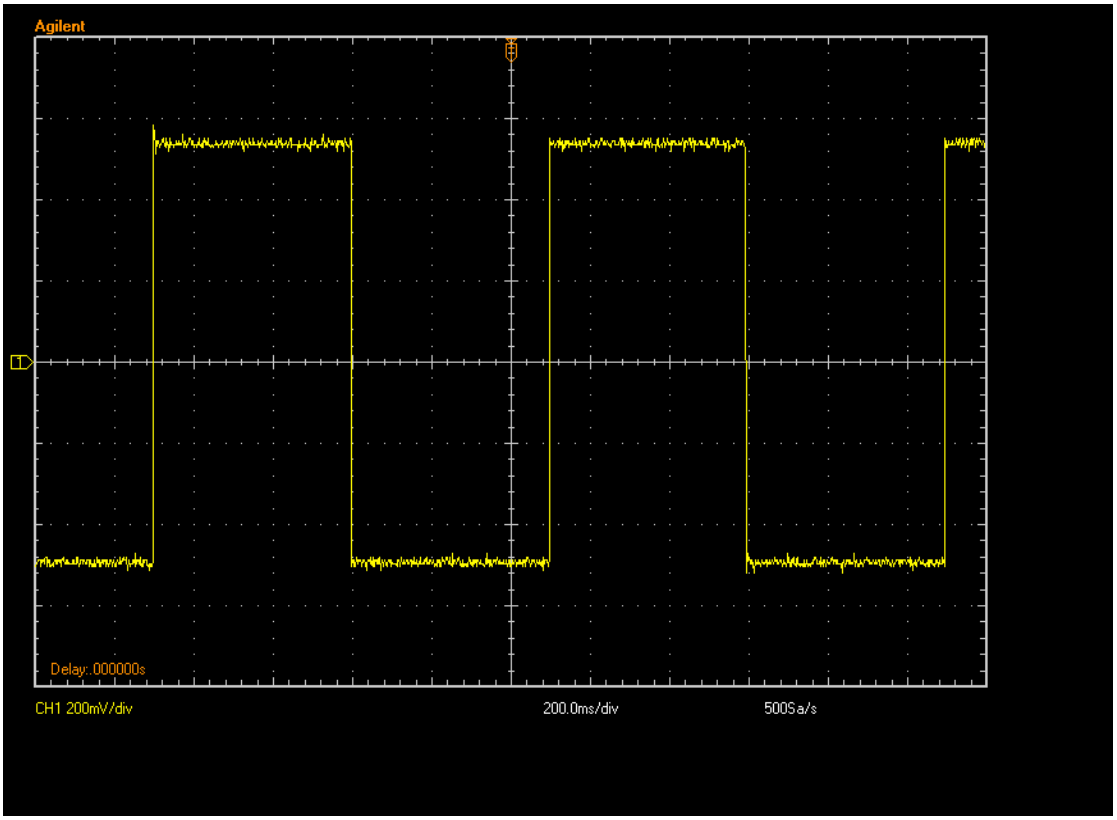


Figura 137 - Osciloscópio comercial [Vpp = 1,09 V; Vrms = 0,517 V; f = 1 Hz]

- Frequência 1 Hz e tensão 126 mV

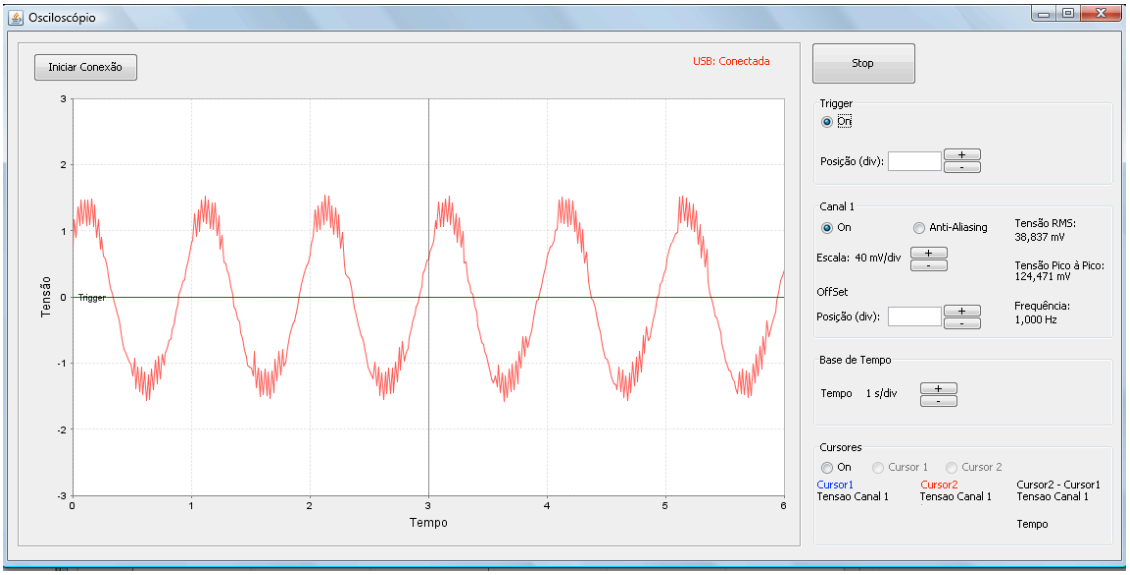


Figura 138 - Osciloscópio projetado [Vpp = 124,471 mV; Vrms = 38,837 mV; f = 1 Hz]

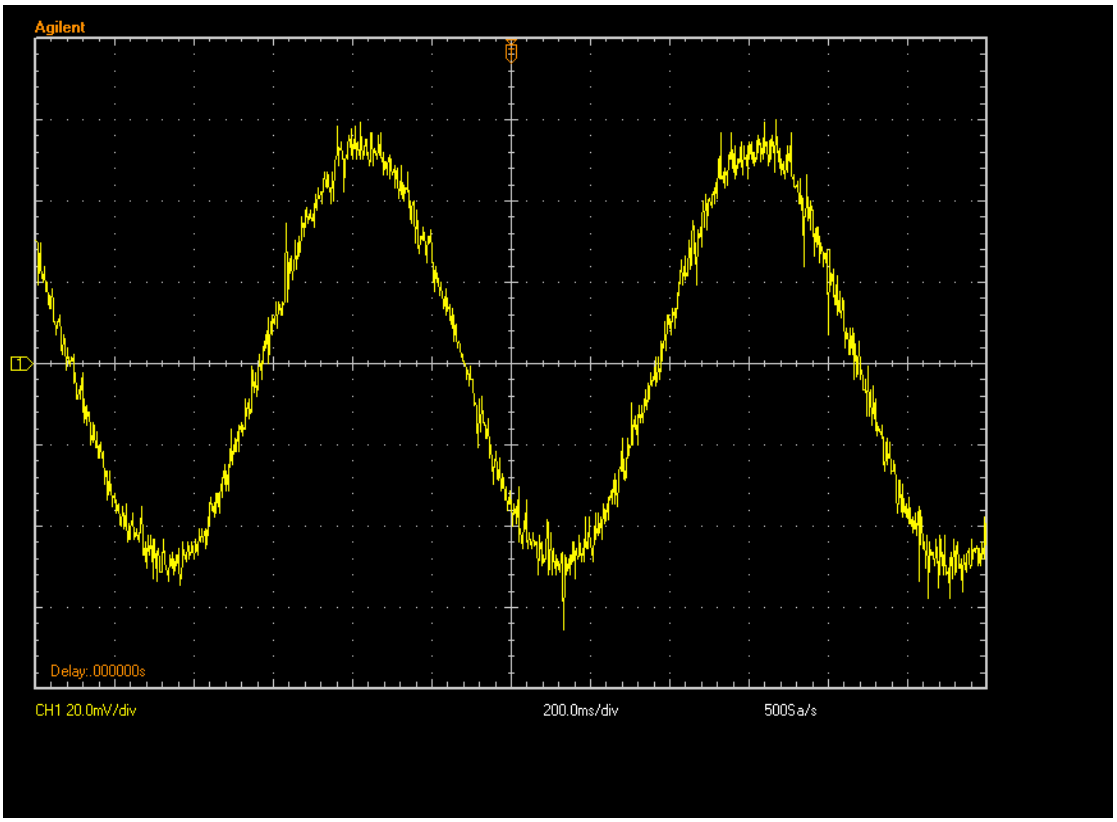


Figura 139 - Osciloscópio comercial [Vpp = 126 mV; Vrms = 36,3 mV; f = 0,994 Hz]

- Frequência 10 kHz e tensão 117 mV

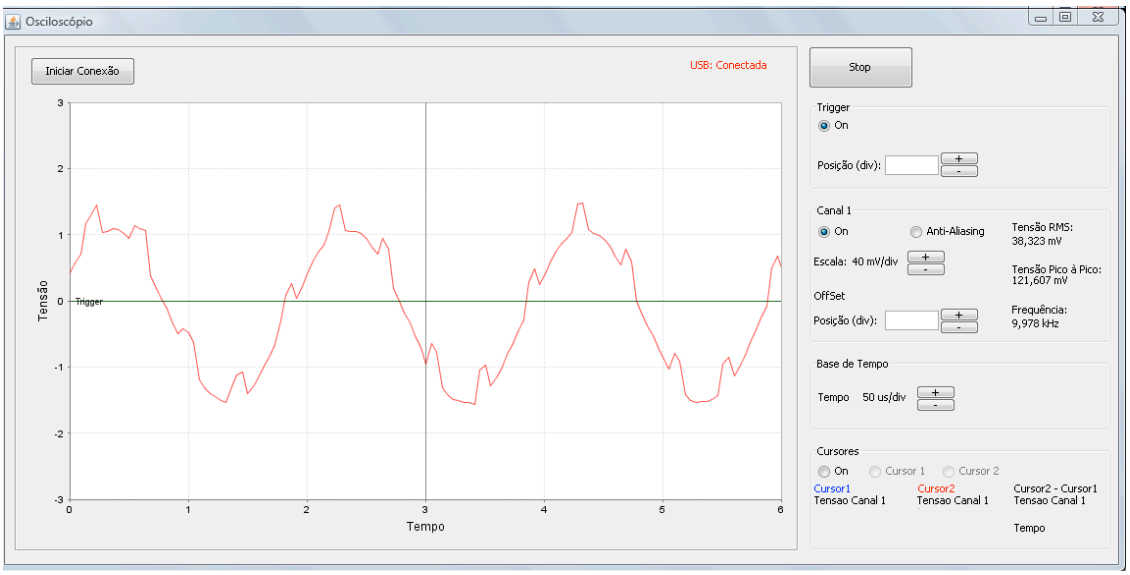


Figura 140 - Osciloscópio projetado [Vpp = 121,607 mV; Vrms = 38,323 mV; f = 9,978 kHz]

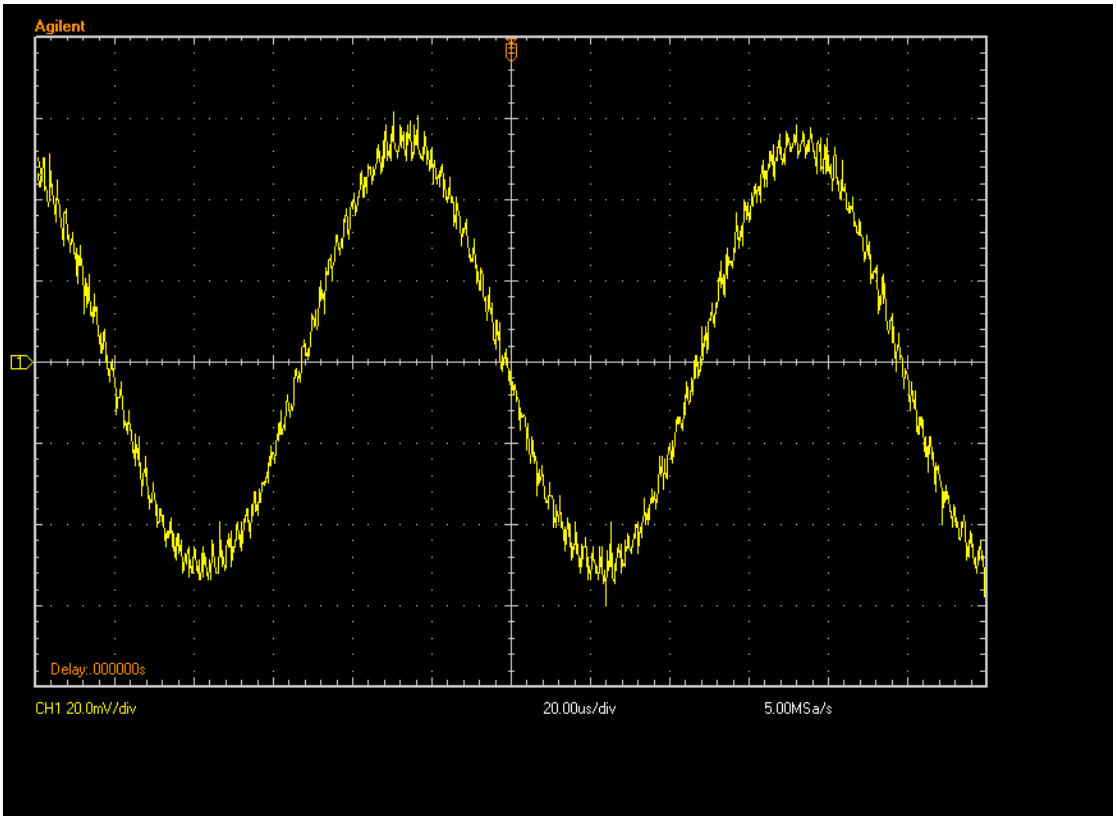


Figura 141 - Osciloscópio comercial [Vpp = 117 mV; Vrms = 36,3 mV; f = 10 kHz]

- Frequência 100 kHz e tensão 112 mV

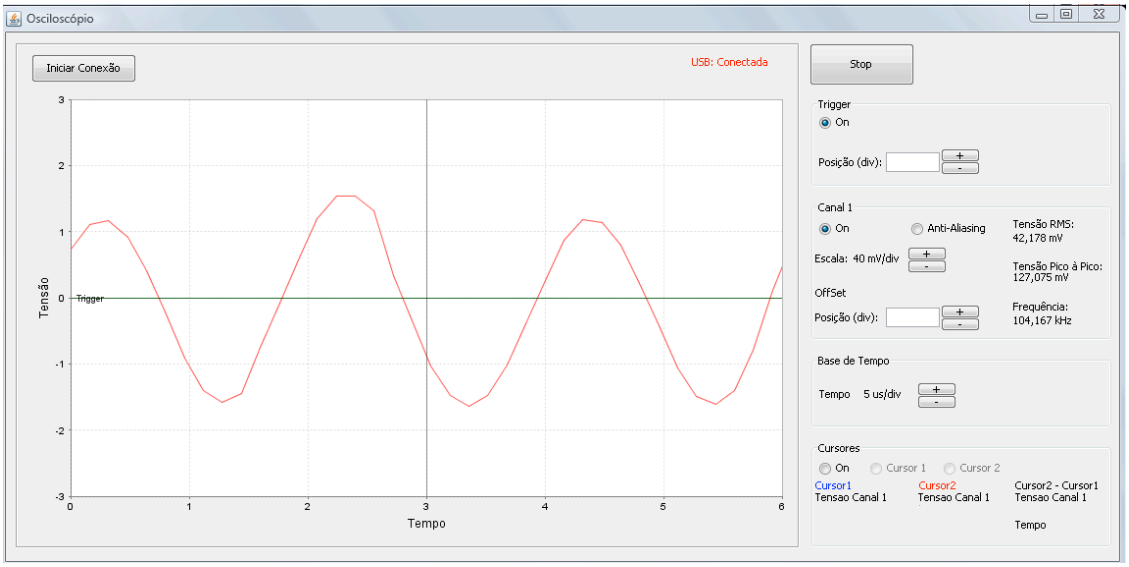


Figura 142 - Osciloscópio projetado [Vpp = 127,075 mV; Vrms = 42,178 mV; f = 104,167 kHz]

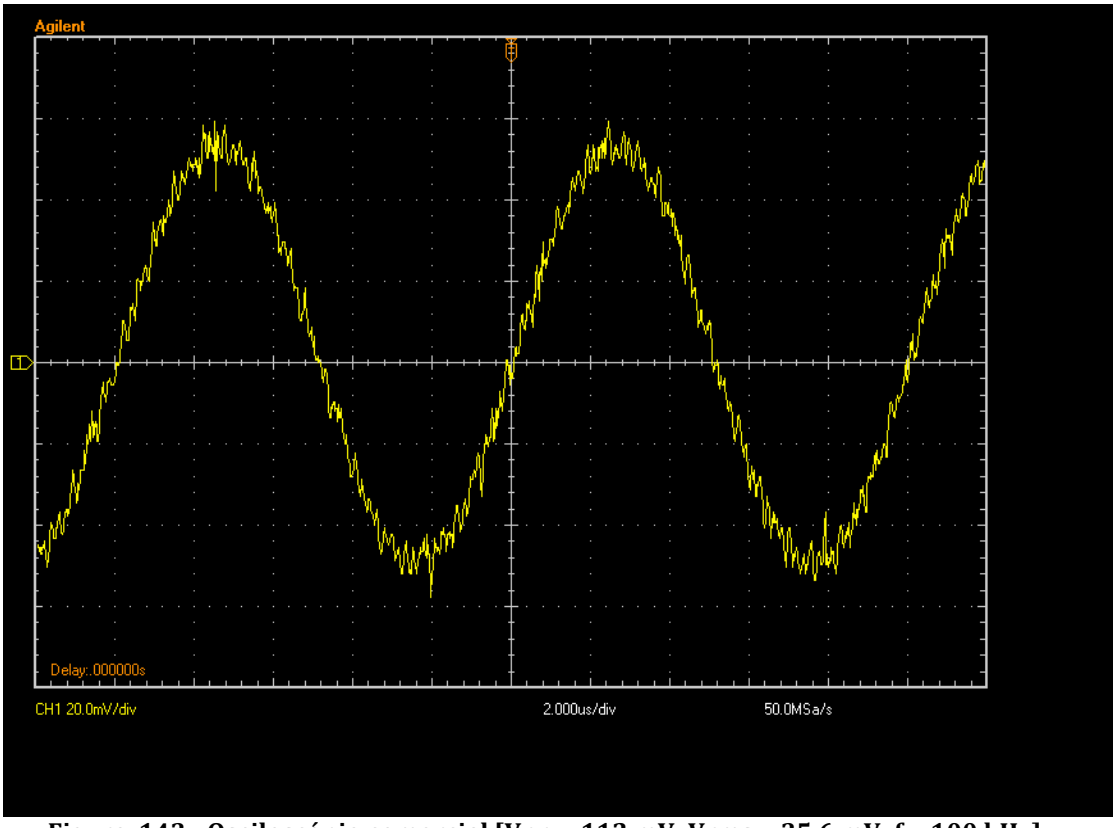


Figura 143 - Osciloscópio comercial [Vpp = 112 mV; Vrms = 35,6 mV; f = 100 kHz]

- Frequência 100 kHz e tensão 119 mV

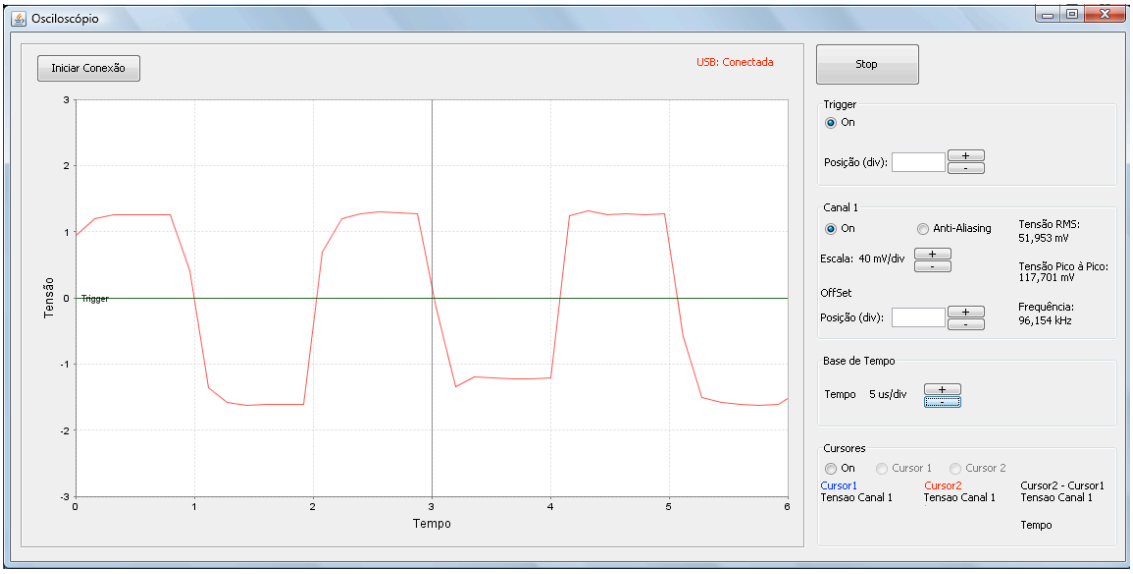


Figura 144 - Osciloscópio projetado [Vpp = 117,701mV; Vrms = 51,953 mV; f = 96,154 kHz]

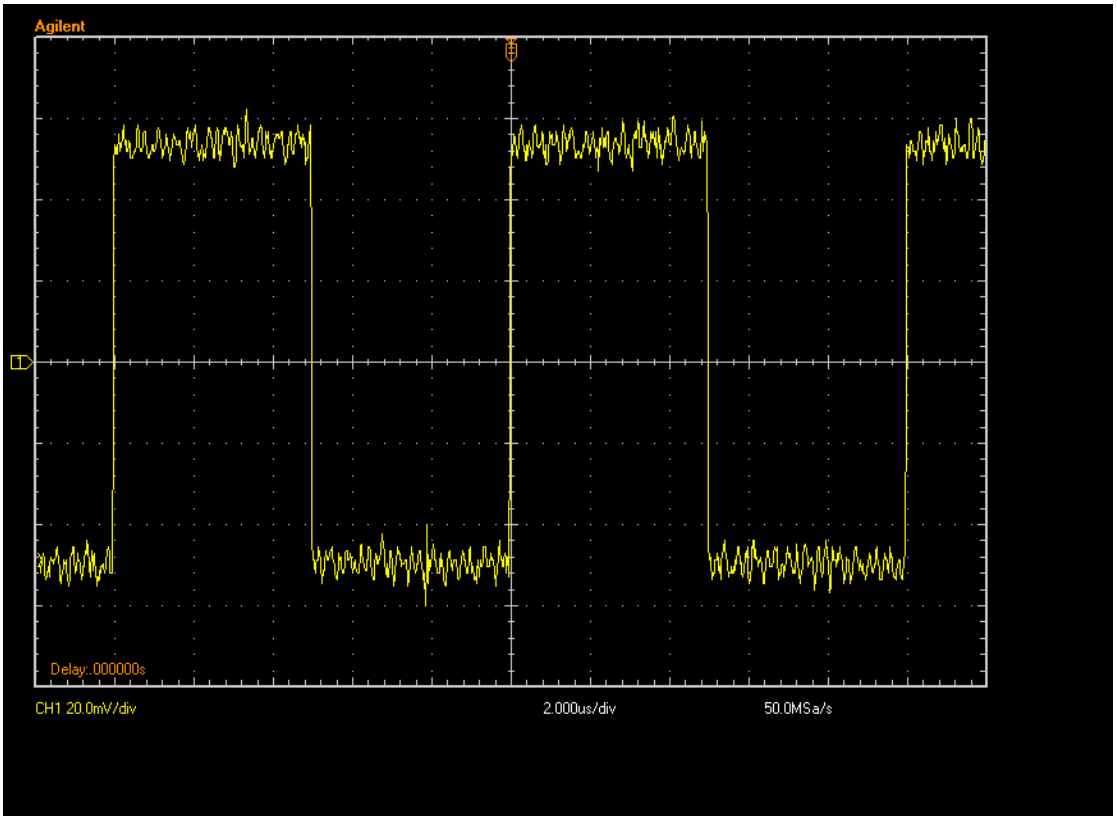


Figura 145 - Osciloscópio comercial [Vpp = 119 mV; Vrms = 51,5 mV; f = 100 kHz]

- Frequência 10 kHz e tensão 121mV

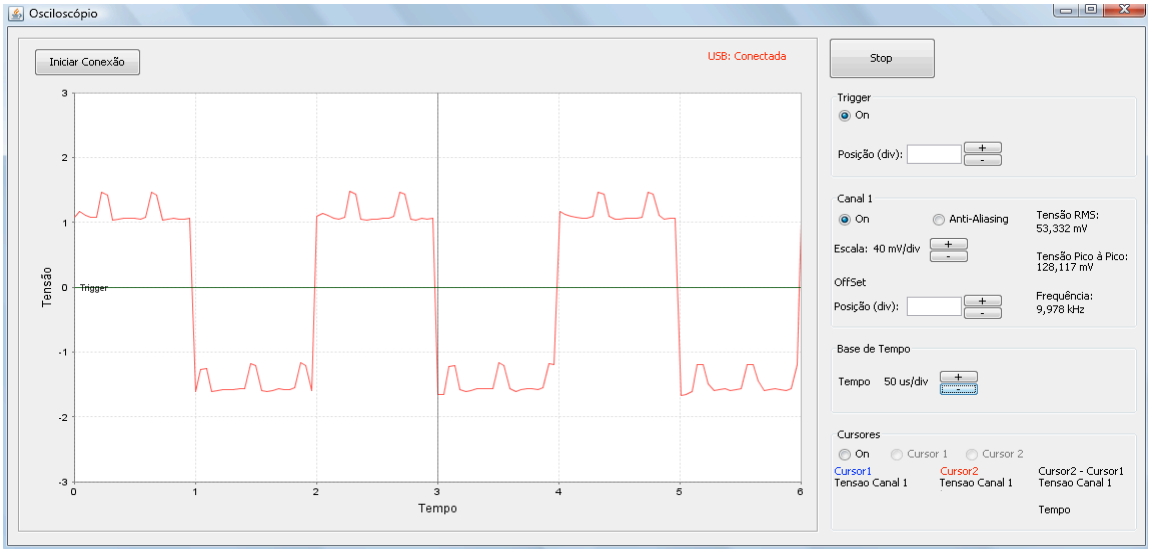


Figura 146 - Osciloscópio projetado [Vpp = 128,117 mV; Vrms = 53,332 mV; f = 9,978 kHz]

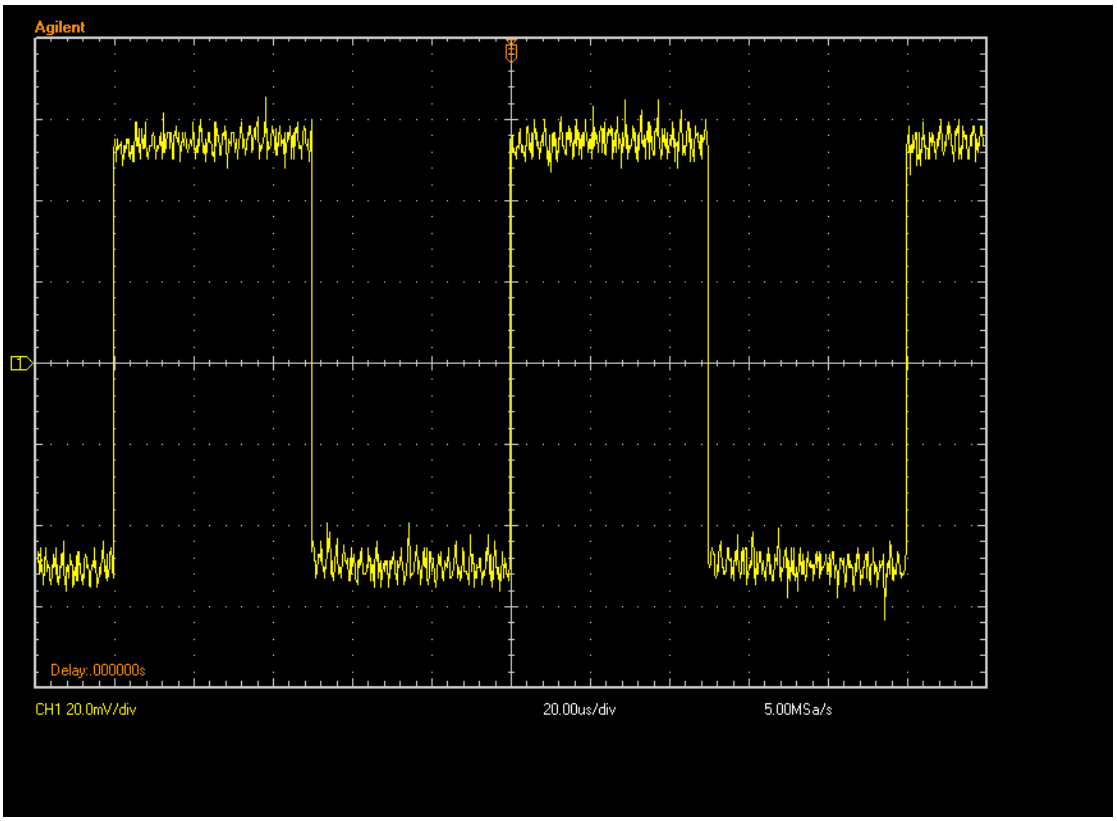


Figura 147 - Osciloscópio comercial [Vpp = 121 mV; Vrms = 52,3 mV; f = 10 kHz]

- Frequência 1 kHz e tensão 123 mV

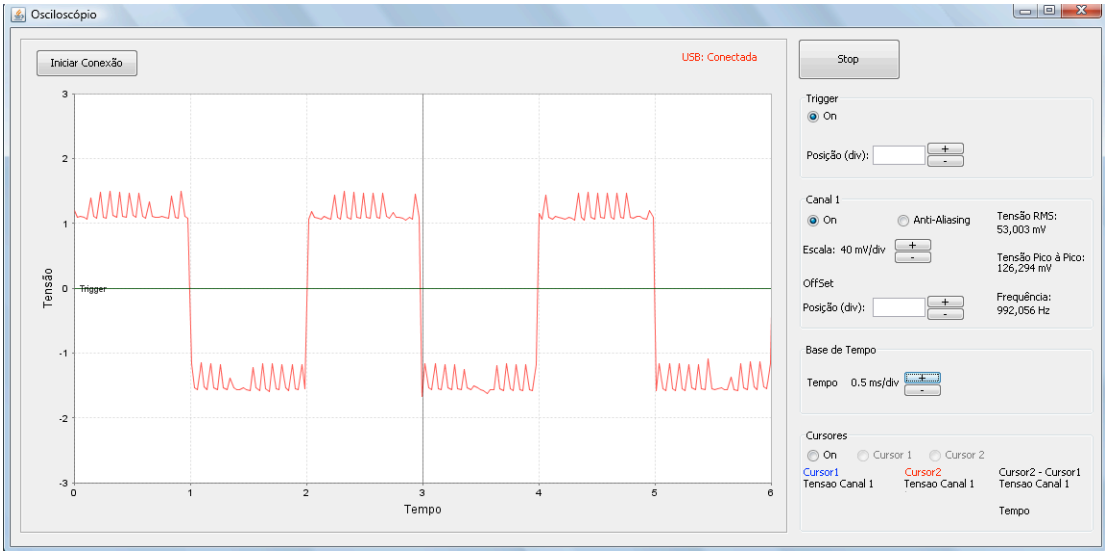


Figura 148 - Osciloscópio projetado [Vpp = 126,294 mV; Vrms = 53,003 mV; f = 992,056 Hz]

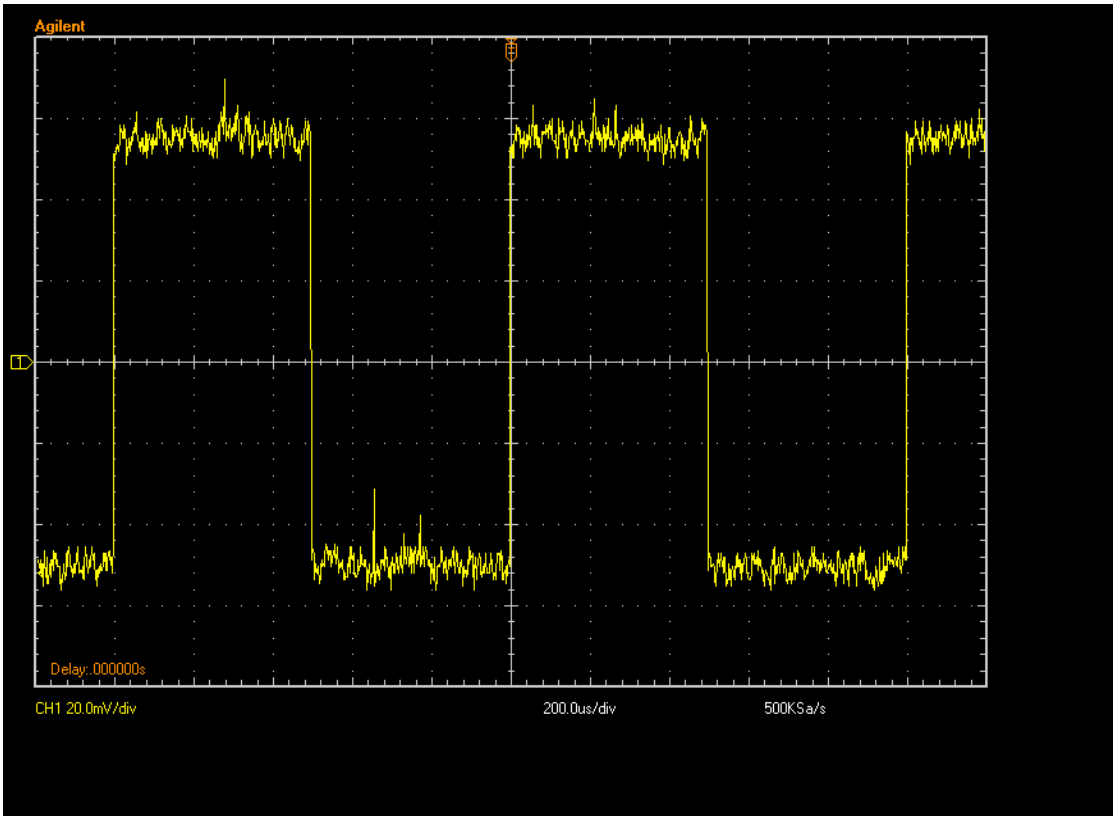


Figura 149 - Osciloscópio comercial [Vpp = 123 mV; Vrms = 52,8 mV; f = 1 kHz]

- Frequência 1 Hz e tensão 122 mV

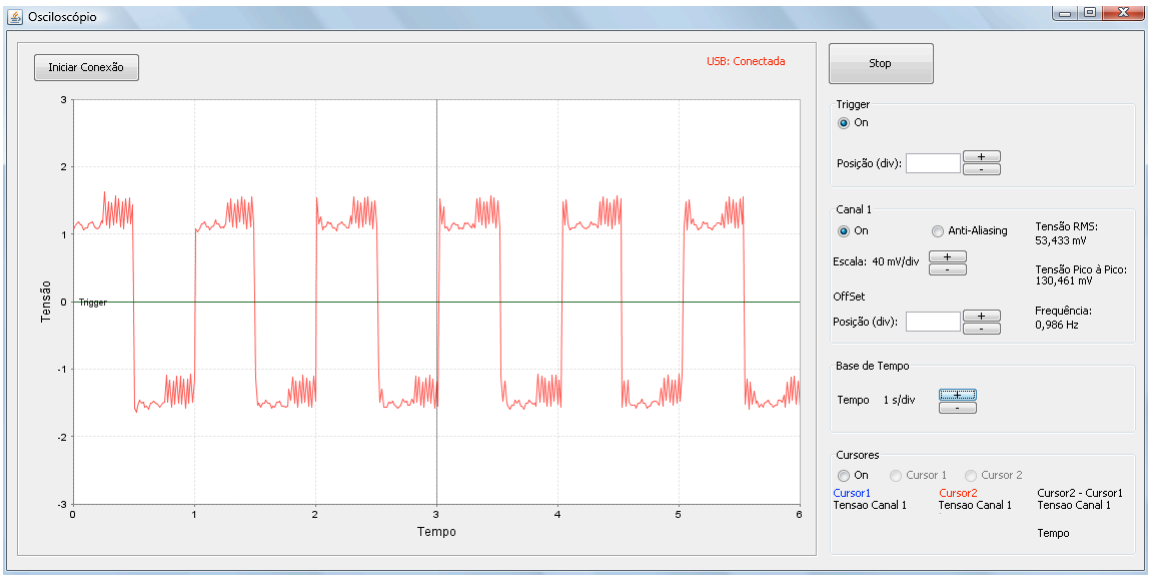


Figura 150 - Osciloscópio projetado [Vpp = 130,461 mV; Vrms = 53,433 mV; f = 0,989 Hz]

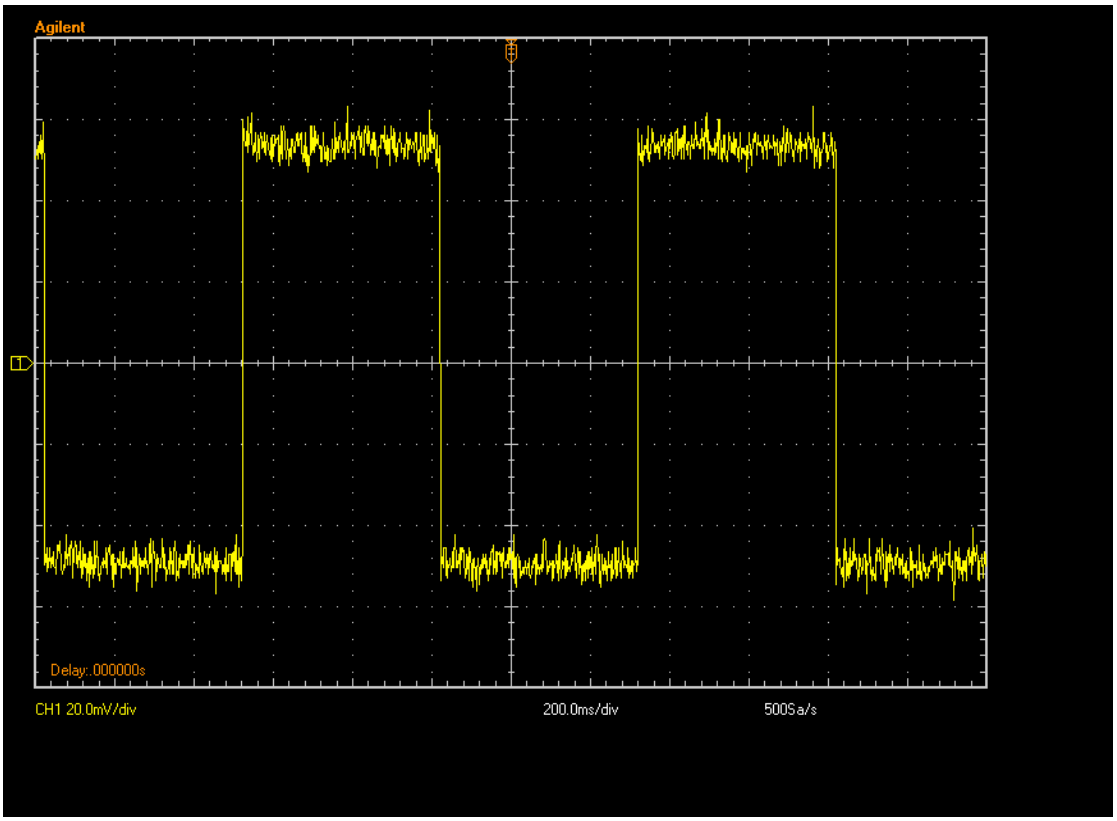


Figura 151 - Osciloscópio comercial [Vpp = 122 mV; Vrms = 51,3 mV; f = 1 Hz]

9.2 MANUAL DO USUÁRIO

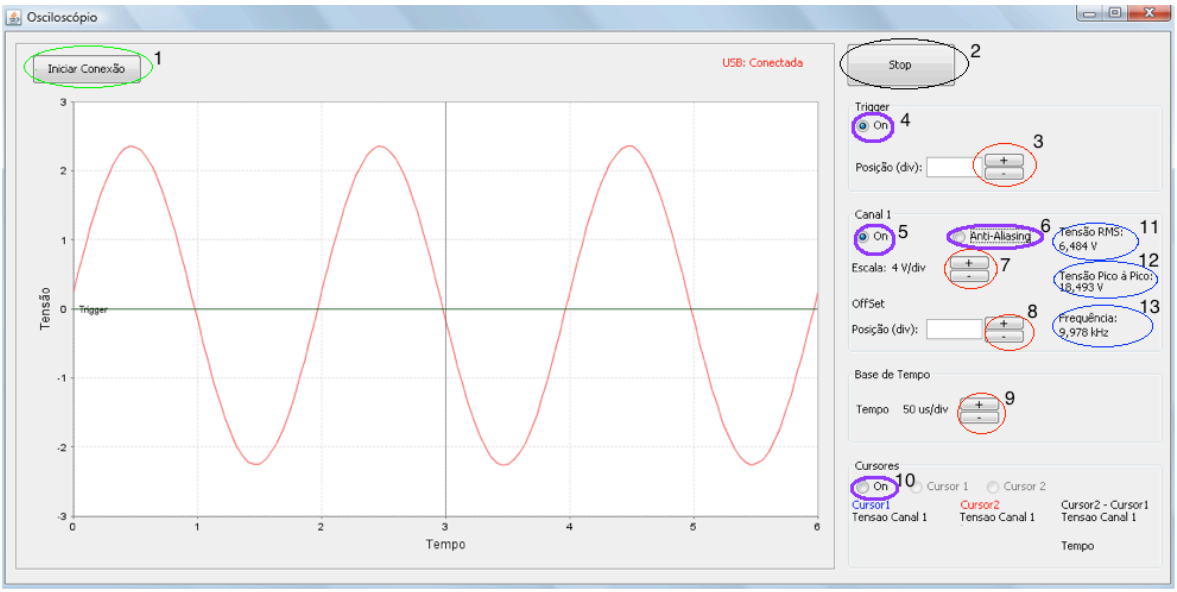


Figura 152 - Software osciloscópio

Este é um manual de operação correspondente ao software do osciloscópio.

1. Primeiramente certifique-se que o sistema de aquisição de sinal está devidamente conectada ao seu computador em uma das portas USB e que tenha instalado as Dlls corretamente (instalação das Dlls no final do manual);
2. Clique em iniciar conexão como mostrado em 1 na Figura 152. Caso a conexão ocorra o status em vermelho é atualizado para USB: Conectada. Caso contrário uma mensagem é mostrada informando que não foi possível se conectar, caso isto ocorra verifique se tudo está devidamente conectado;
3. Para ativar o canal é necessário clicar na opção “On” em canal 1 a qual está referenciada pelo número 5 da Figura 152, para desativar clique novamente sobre ele;
4. Caso o sinal não apareça corretamente em sua tela, é necessário mudar as escalas conforme indicado nos números 7 e 9 da Figura 152;
5. Os valores de tensão RMS, tensão Pico à Pico e Frequência são informas nos números 11, 12 e 13 da Figura 152;

6. Caso queira “congelar” a forma de onda mostrada em seu visor, clique no botão “Stop” (número 2 da Figura 152);
7. Para ativar o trigger é necessário clicar na opção “On” em trigger a qual está referenciada pelo número 4 da Figura 152, para desativar clique novamente sobre ele;
8. Para alterar o trigger use os botões de ”+” e “-“ referenciado pelo número 3 da Figura 152;
9. Para ativar o Ant-Aliasing é necessário clicar na opção “On” em canal 1 a qual está referenciada pelo número 6 da Figura 152, para desativar clique novamente sobre ele;
10. Para ativar os cursores é necessário clicar na opção “On” em cursores a qual está referenciada pelo número 10 da Figura 152, para desativar clique novamente sobre ele;

Instalação das DLLs

-Copie o arquivo win32com.dll para o diretório C:\JavaSDK\BIN (isto È, o diretório onde o J2SDK foi instalado no seu PC).

- Copie o arquivo comm.jar para o diretÛrio C:\JavaSDK\BIN\LIB. (n„o È essencial)

-Copie o arquivo javax.comm.properties para o diretório C:\JavaSDK\BIN\LIB.

10 Referências

[1] - DSO Nano - Open source Digital Oscilloscope based on STM32. Disponível em <http://code.google.com/p/dsonano>. Acesso em 09 de setembro de 2011;

[2] - PC Oscilloscope and Data Acquisiton Products. Disponível em <http://www.picotech.com/oscilloscope.html>. Acesso em 09 de setembro de 2011;

[3] - Bitscop Pc Oscilloscopes And Analyzers. Disponível em <http://www.bitscope.com/>. Acesso em 09 de setembro de 2011;

[4] – Tecktronix. Disponível em <http://www.tek.com/>. Acesso em 10 de novembro de 2011.

[5] Tektronix, “XYZs of Oscilloscopes”, versão de 1995, atualmente em nova edição disponível em <www.tek.com>. Acesso em 20 de novembro de 2011.

[6] ABC do Osciloscópio – Disponível em <http://ave.dee.isep.ipp.pt/~mjf/PubDid/ABC_Osc.PDF>. Acesso em 1 de dezembro de 2011.

[7] – JfreeChart - Disponível em <http://www.jfree.org/jfreechart/>. Acesso em 20 de novembro de 2011.

[8] – Wikipédia - Valor Eficaz - Disponível em <http://pt.wikipedia.org/wiki/Valor_eficaz>. Acesso em 20 de novembro de 2011.

[9] - Altium – Disponível em <www.altium.com>. Acesso em 10 de setembro de 2011;

[10] – Oracle – Disponível em <www.oracle.com/technetwork/java/index-jsp-141752.html>. Acesso em 20 de novembro de 2011.

[11] STMicroeletronics. Datasheet: Medium-density performance line ARM-based 32-bit MCU. Electronic Publication, 2011.a